

## 第1章

## 組み込みシステム概論

いろいろな機械や機器に組み込まれて、その制御を行うコンピュータシステムのことを、組み込みシステム (embedded system) と呼ぶ。本章では、組み込みシステムとはどのようなシステムであるかを説明し、組み込みシステムの特性やそのソフトウェア (組み込みソフトウェア) 開発の特徴について述べる。また、組み込みソフトウェア開発の最近の動向についても紹介する。

## 1.1 組み込みシステムとは

組み込みシステムの適用分野と適用した機器の例を表1.1に挙げる。組み込みシステムとは、「機械や機器に組み込まれて、その制御を行うコンピュータシステム」のことだと述べたが、この定義には曖昧なところがある。実際、どのようなシステムのことを組み込みシステムと呼ぶのかは、人によって異なっている。

たとえば、最近のプラント制御システムには、パーソナルコンピュータ(PC)ベースのハードウェアが用いられる場合も多い。この場合、ハードウェア的にはPCそのものであり、「組み込まれて」いるとは考えにくい。そのため、プラント制御システムは組み込みシステムではないと考える人も多い。しかし、工場という大規模な「機械」に組み込まれて、それを制御しているコンピュータシステムであると考えれば、上の定義にあてはまることになる。実際、プラント制御システムには、次節で述べる組み込みシステムの特性の多くがあてはまる。

別の例として、ゲームマシンが挙げられる。ゲームマシンの形態によっては、中にコンピュータが組み込まれていると考えるのが自然なものもあれば、コンピュータそのものであると考えたほうが自然なものもある。最近のゲームマシンには、PCを小型化・簡略化して作られているものもあり、その場合には、ハードウェア構成やソ

〔表1.1〕  
組み込みシステムの  
適用分野と機器の例

適用分野		利用例
家電機器		電子レンジ, 炊飯器, 冷蔵庫, 洗濯機, 乾燥機, エアコン
AV機器		テレビ, ビデオ, デジタルカメラ, オーディオ機器
娯楽 / 教育機器		ゲームマシン, 電子楽器, カラオケ, パチンコ
個人用情報機器		PDA, 電子手帳, カーナビ
パソコン周辺機器		プリンタ, スキャナ, ディスクドライブ, DVDドライブ
OA機器		コピー, FAX
通信機器	端末	携帯電話機, 留守番電話機
	ネットワーク設備	ネットワークルータ, ハブ, 携帯電話基地局, PBX, 交換機
運輸機器		自動車, 信号機, 鉄道車両, 鉄道制御, 航空機, 船舶
工業制御 / FA機器		プラント制御, NC工作機械, 工業用ロボット
設備機器		ビル用照明, ビル用空調, ビル用電力システム, エレベータ
医用機器 / 福祉機器		血圧計, 心電計, レントゲン, CTスキャナ
宇宙 / 軍事		ロケット, 人工衛星, ミサイル
その他の業務用機器		業務用データ端末, POS端末, 自動販売機
その他の計測機器		シンクロスコープ, ICテスタ, 電力メータ

ソフトウェア構成の面でもPCに近いものになっている。

組み込みシステムを、上のような例まで含めて広くとらえると、PCやワークステーションなどのいわゆる汎用システムを除くほとんどのコンピュータシステムが、組み込みシステムであるということになる。汎用システム以外であることを言い換えると、ある応用に専用化されたコンピュータシステムを、組み込みシステムと呼んでいることになる。

上に例で挙げたゲームマシンは、ゲームという応用に専用化されているとも捉えられるし、多種類のゲームができるようになるという意味では、汎用という捉え方もできる。このことから、ゲームマシンは、組み込みシステムと汎用システムの中間的なものと考えるのがよいだろう。実際、ゲームマシンに関しては、次節で述べる組み込みシステムの特性で、あてはまらないものも多い。また、PDAについても同様のことがいえる。

組み込みソフトウェアとは、一般には組み込みシステムのソフトウェアのことをいうが、「機械や機器に固定されていて変更されることのないソフトウェア」とする定義もわかりやすい<sup>注1</sup>。後者の定義では、一つのシステムを構成するソフトウェアも、組み込みソフトウェアとそうでない部分に分けることができる。たとえば、PCのBIOSは組み込みソフトウェアということになる。

## 1.2 組み込みシステムの特性

組み込みシステムは、制御対象である機械や機器の多様性を反映して、極めて多様である。システムの規模の面では、8ビットのワンチップマイコンを用いた小規模なシステムから、32ビット、さらには64ビットのプロセッサを複数用いた大規模なシステムまで含まれる。また、システムに対する要求事項の面でも、自動車や航空機の制御システムのように高い信頼性とリアルタイム性の確保が最重要課題となるシステムもあれば、家電製品やAV機器のようにコストダウン要求が厳しいシステムもある。

そのため、すべての組み込みシステムに共通する特性はないといっても過言ではない。とはいえ、多くの組み込みシステムに共通する特性はある(そうでなければ、「組み込みシステム向け」のソフトウェア開発技術は存在しえない!)。以下で述べる四つの特性は、特に多くの組み込みシステムに共通するものである。

### 1.2.1 専用化されたシステム

先に述べたとおり、組み込みシステムをもっとも広く捉えると、ある応用に専用化されたコンピュータシステムということになり、この特性は組み込みシステムの定義であるということもできる。

ある応用に専用化されたシステムという特性は、システムを最適に設計するためには有利な特性である。応用が限定されていることから、ハードウェアやOSの使われ方も限定される。そのため、ハードウェアやOSは、アプリケーションプログラムが使う機能のみを備えていればよい。また、頻繁に使う機能や高速化したいプログラムで使う機能を高速に処理できるように、ハードウェアやOSを最適化することもできる。

### 1.2.2 厳しいリソース制約

プロセッサの処理能力やメモリ容量などの計算リソースに対して、厳しい制約が課されているシステムが多い。言い換えると、処理能力の低いプロセッサを使うことや、使用するメモリ量を少なくすることが強く求められる。

このような要求がシステムのコストダウンの要求からくることは理解しやすいだろう。大量生産される機器に組み込まれるシステムの場合、部品のコストを少しでも下げることができれば、全体では大きなコストダウンにつながる。

簡単な計算をしてみる。組み込みシステムの中には、携帯電話のように一つの機種で100万台以上も生産するものもある。このようなシステムでは、100円安いプロセッサを使うことができれば、全体では1億円以上のコス

注1：これをファームウェアと呼ぶ場合もあるが、ファームウェアという言葉は違う意味で使われることもあるので注意が必要である。

トダウンになる。

それに加えて、低消費電力化に対する要求や小型化・軽量化の要求、さらには動作温度条件やEMCといった動作環境から、厳しいリソース制約が生じる場合もある。とくに、携帯機器では消費電力を下げるのが極めて重要で、そのためには、プロセッサの動作クロック周波数をできる限り低くするほうが有利である。

### 1.2.3 高い信頼性

機械や機器を制御することから、組み込みシステムには高い信頼性が求められるのが通常である。プラントや自動車の制御といった応用では、システムの誤動作が人命にかかわる事態につながるため、システムには極めて高い信頼性が要求される。家電製品などの民生機器の場合でも、システムの誤動作が重大な事態を引き起こす可能性を否定できない。

また、大量生産される機器の場合には、システムに問題があった場合には、一度販売した機器を回収(リコール)してシステムの修正を行う必要がある。回収には高いコストがかかることに加えて、メーカーのブランドに対する信頼が損なわれるおそれもあるため、機器の出荷前に徹底的にシステム検証を行うことが求められる。

さらに、技術的な要因ではないが、ユーザーが機械や機器の信頼性を期待している(それに対して、PCの信頼性は低いものとあきらめている)ことも、組み込みシステムが高い信頼性を求められる理由の一つである。

### 1.2.4 リアルタイム性

リアルタイム性とは、単に計算処理速度が速いことやレスポンス時間が短いことをいうのではなく、システムが定められた時間要件を満たして動作する性質をいう。多くの組み込みシステムは、やはり機械や機器を制御することから、制御する対象の機械や機器によって定まる時間条件を満たして動作しなければならない。つまり、リアルタイム性が求められることになる。

ここに述べたこと以外にも、一部の組み込みシステムにあてはまる特性をいくつか挙げるができる。たとえば、組み込みシステムの中には、極めて寿命の長いシステムがある。家電製品であっても10年程度は使用するし、プラント制御や公共系のシステムでは、寿命が20年というものも珍しくはない。その一方で、寿命の短いシステムもあり、汎用コンピュータと比べて特徴的ではあるものの、多くの組み込みシステムに共通とはいえない。また、人工衛星の制御システムのように、実稼働が始まると保守ができないが、できたとしても極めて難しいシステムもある。

以上で挙げた特性が、組み込みシステムのハードウェアおよびソフトウェアの設計・開発を特徴づけている。次節では、組み込みソフトウェア開発の特徴について見ていく。

## 1.3 組み込みソフトウェア開発の特徴

ここでは、組み込みシステムのソフトウェア開発に見られる、汎用システムにはない特徴について整理する。これらの特徴の多くは、前節で述べた組み込みシステムの特性と関係している。繰り返しになるが、これらの特徴があてはまらないシステムもある。

### 1.3.1 ハードウェアに密着したプログラミング

組み込みシステムのハードウェア構成は、制御する対象の機械や機器によって異なる。機械や機器の種類に応じて、各種のセンサやアクチュエータ、通信インターフェースをもっており、周辺デバイスも特殊なものが多い。

そのため、ソフトウェア開発の中で、デバイスドライバなど、ハードウェアを直接扱う部分の比率が大きい。ハードウェアを直接扱うソフトウェアの開発は、ハードウェアに関する知識が必要であることに加えて、動作がタイミングに依存する場合も多く、開発が難しいといわれている。そのため、ノウハウに負う部分の大きいのが現状である。

〔図1.1〕  
リモート開発環境の例



### 1.3.2 開発環境とターゲットシステムの分離

組み込みソフトウェアの開発においては、ソフトウェアを動作させるターゲットシステムがソフトウェア開発に必要な機能をもっているとは限らず、開発環境を動作させるホストシステムとターゲットシステムが分離しているのが通常である(図1.1)。そのため、クロスコンパイラやクロスアセンブラ、リモートデバッガ、ICE( In-Circuit Emulator )などの組み込みソフトウェア開発に独特な開発ツールが用いられる。これらの開発ツールについては、第3章で詳しく述べる。

リアルタイム性を必要とするシステムの中には、システムの検証やデバッグ中であっても、システムを停止させてはならないものがある。また、デバッガを使うことで実行時間が変わることが許されない場合もある。上で挙げたツールの中で、ICEはこのようなシステムの検証・デバッグを支援するためのツールとしても重要である。最近では、チップの外部にバスが出ていないことが多いため、ICEが使えない場面が増えている。ICEを代替する機能として、多くのプロセッサがオンチップデバッグ機能を備えるようになってきた。

### 1.3.3 ハードウェアとの並行開発とコ・デザインの可能性

組み込みシステムでは、そのシステム専用にハードウェアが開発されることが多いが、その場合、ハードウェアが完成する前にソフトウェアの開発を始める並行開発が行われるのが一般的である。ハードウェアとソフトウェアの並行開発は、システムの開発期間を短縮するために有効である。

さらに、開発期間を短縮するためには、ハードウェアが完成する前にソフトウェアの検証・デバッグができることが望ましい。そのために、シミュレータを用いてソフトウェアの検証やデバッグを行うことも多い。また、設計したハードウェアとソフトウェアを一体でシミュレーションするコ・シミュレーション(協調シミュレーション)ツールが使われることもある。さらに、ソフトウェアを効率的に実行するために求められるハードウェアの要件を明確にし、それにしたがってハードウェアを設計するコ・デザイン(協調設計)技術が適用されることも多い。

ハードウェアとソフトウェアを並行開発する場合には、ハードウェアは正しく動作するという前提でソフトウェアのデバッグを行うことはできない。問題が生じた場合には、ソフトウェアに原因があるのかハードウェアの問題なのかを切り分けることが必要になる。さらにいうと、開発環境(コンパイラやデバッガ)やOSも新規に開発されたものである場合も多く、問題の切り分けにはそれぞれの技術に関する広範な知識をもった技術者が求められる。

### 1.3.4 多様なハードウェアと多様なOS

組み込みシステムは、規模の面でも要求事項の面でも多様であることから、用いられるプロセッサやOSも極めて多様である。

そのため、ソフトウェアの再利用性を上げようとする、プロセッサやOSに依存しないようにすることが重要である。たとえば、組み込みシステム用のリアルタイムOS(RTOS)の多くは、プロセッサの違いを隠蔽するための仕組み(HAL:ハードウェア抽象化レイヤ)の上にプロセッサに依存しない部分を載せる形で実現されているし、RTOSの上で動作するソフトウェア部品は、RTOSの違いを隠蔽するための仕組みをもっていることが多い。

さらに、DSP(Digital Signal Processor)やメディアプロセッサなどの応用に特化したプロセッサが用いられる

ことも多いが、このようなプロセッサにはコンパイラがないが、あったとしても生成されるコードの効率が悪い場合が多く、いまだにアセンブリ言語でのソフトウェア開発が必要な場面も多い。

### 1.3.5 システム内のソフトウェアは信頼できるという前提

組み込みソフトウェアは、機械や機器に固定されて提供され、それを制御することのみを目的に開発されるのが通常である。開発者は、システム内でどのようなソフトウェアが動作しているかを把握しており、システムの動作を妨害するようなソフトウェアが組み込まれるおそれはない。そのため、ソフトウェアの検証が完了すれば、システム内のソフトウェアは信頼できるという前提が成り立つ。

組み込みシステム用のRTOSの多くが保護のための機能をもっていないのは、保護機能を実現するためのオーバヘッドが大きいことに加えて、この前提が成り立っているためである。

## 1.4 組み込みシステムの簡単な歴史と現状

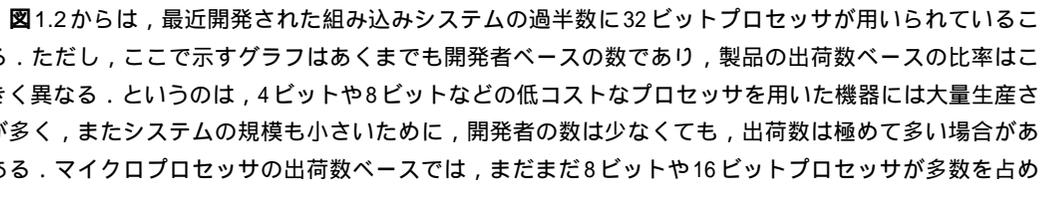
コンピュータが高価であった時代には、(広い意味での)組み込みシステムの適用分野も、高価な機械の制御に限定されていた。具体的には、プラント制御などの工業制御の分野から始まって、大規模な交通制御(鉄道制御や航空管制)や通信システムなどへと徐々に広がっていった。

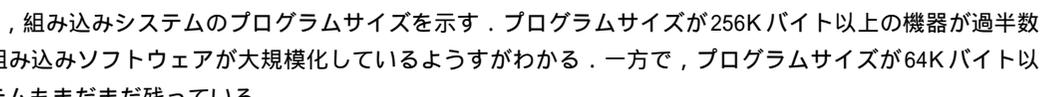
組み込みシステムの適用分野が急速に広がったのは、1970年代の初めにマイクロプロセッサが発明されて以降である。最初のマイクロプロセッサである4004は電卓に使うために開発されたので、広い意味での組み込みシステム用途であったということが出来る。低コスト化が可能なマイクロプロセッサの登場により、組み込みシステムの適用分野は急速に拡大していった。

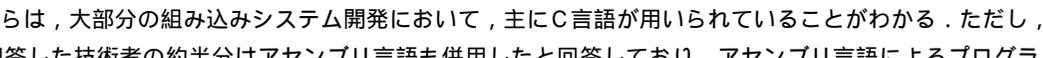
1980年代の初めころにはマイクロプロセッサの低コスト化が進み、家電製品などの民生機器分野にまで、組み込みシステムが使われるようになった。ただし、民生機器分野で使われたのは、処理能力の低い4ビットや8ビットのプロセッサがほとんどであった。マイクロプロセッサで動作するRTOSが登場したのはこの頃である。

1990年代半ばころからは、PCやインターネットなどの情報インフラの普及と、半導体技術の進歩により高性能なマイクロプロセッサが低コストで利用できるようになったことから、機械や機器のネットワーク化・デジタル化が急速に進展した。ネットワーク化やデジタル化により、組み込みシステムに求められる処理能力は一段と大きくなり、その役割を一層大きいものとした。また、そのソフトウェアも急速に大規模化・複雑化している。

以下では、2000年度にトロン協会が行ったアンケート調査結果<sup>注2</sup>に基づいて、組み込みシステム開発の現状を見ていく。この調査結果は、国内の組み込みシステム技術者を中心に約600名からの回答を集計したものであり、最近開発された組み込みシステムにどのような技術が適用されているかを概観することができる。

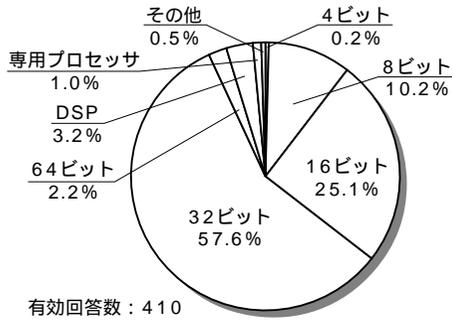
最初に、1.2からは、最近開発された組み込みシステムの過半数に32ビットプロセッサが用いられていることがわかる。ただし、ここで示すグラフはあくまでも開発者ベースの数であり、製品の出荷数ベースの比率はこれとは大きく異なる。というのは、4ビットや8ビットなどの低コストなプロセッサを用いた機器には大量生産されるものが多く、またシステムの規模も小さいために、開発者の数は少なくとも、出荷数は極めて多い場合があるためである。マイクロプロセッサの出荷数ベースでは、まだまだ8ビットや16ビットプロセッサが多数を占めている。

1.3は、組み込みシステムのプログラムサイズを示す。プログラムサイズが256Kバイト以上の機器が過半数を占め、組み込みソフトウェアが大規模化しているようすがわかる。一方で、プログラムサイズが64Kバイト以下のシステムもまだまだ残っている。

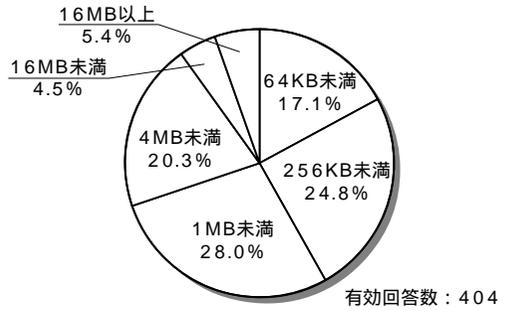
1.4からは、大部分の組み込みシステム開発において、主にC言語が用いられていることがわかる。ただし、C言語と回答した技術者の約半分はアセンブリ言語も併用したと回答しており、アセンブリ言語によるプログラ

注2：元のデータは、<http://www.ert1.jp/TRON/>にある。

〔図1.2〕最近開発した組み込みシステムに用いたプロセッサの種類別 (トロン協会のホームページより)



〔図1.3〕最近開発した組み込みシステムのプログラムサイズ (トロン協会のホームページより)



ムを完全に回避することは難しいのが現状である。C++やJavaなどのオブジェクト指向は、比率的には大きいものの、徐々に利用が広がりつつある。

組み込んだOSの比率を図1.5に示す。このグラフからは、大部分の組み込みシステムにOSが使われていることがわかる。また、使われたOSのほとんどは、リアルタイムOSに分類されるものである。

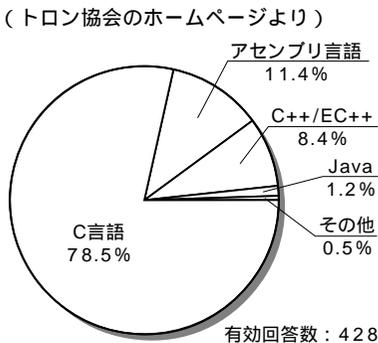
一方で、使われたOSは多種多様であり、どのOSのシェアも10%に満たない。これは、開発するシステムや開発現場の要求事項に合わせて、最適なOSが選ばれているためと考えられる。また、このグラフには正確な数値は示されていないが、全体の約4割のシステムでITRON仕様に準拠したOSが利用されており、ITRON仕様がデファクト標準仕様として定着していることがわかる。

OSの利用状況をより詳しくみるために、プロセッサ別やプログラムサイズ別にOSの利用率を集計したものを図1.6と図1.7に示す。これらのグラフからは、システムの大規模化とともにOSの利用率が上がることが確認できるが、組み込みシステム開発技術者には、自分の開発するシステムにOSを組み込むべきかどうかの目安として役立つものと思う。

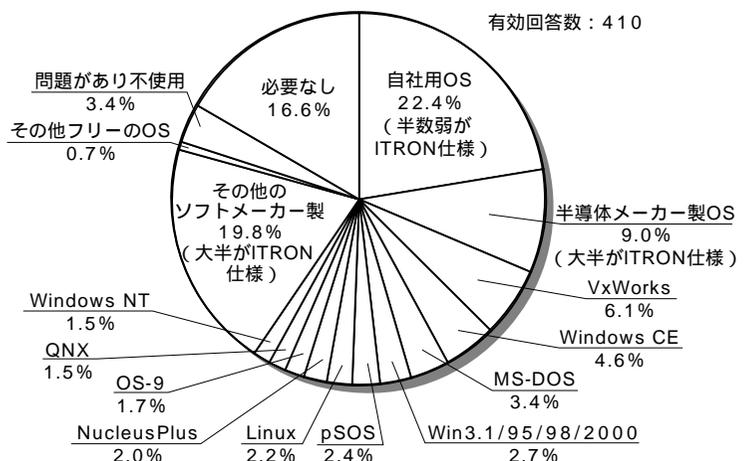
## 1.5 組み込みシステムの適用例

本節では、組み込みシステムの実際の適用例として、自動車に使われている組み込みシステムの概要と、その

〔図1.4〕最近開発した組み込みシステムに用いたプログラミング言語 (トロン協会のホームページより)

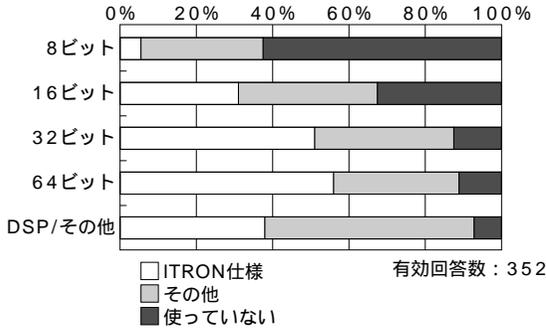


〔図1.5〕最近開発した組み込みシステムに用いたOS (トロン協会のホームページより)



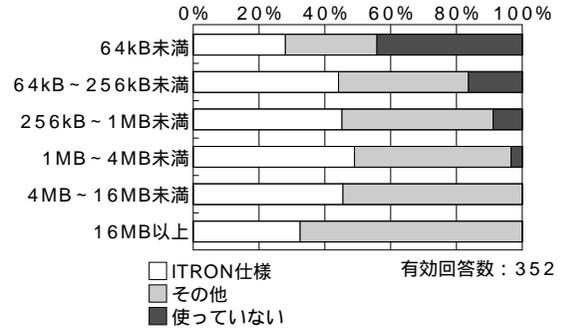
〔図1.6〕 プロセッサ別のOS利用率

(トロン協会のホームページより)



〔図1.7〕 プログラムサイズ別のOS利用率

(トロン協会のホームページより)



中の一つであるエンジン制御システムについて紹介する。また、自動車内のネットワークについても紹介する。自動車は、身近なシステムであり、前に述べた組み込みシステムに特徴的な性質を備えている。

### 1.5.1 自動車内の組み込みシステムの特徴

1台の自動車の中に使われているコンピュータ(または、マイクロプロセッサ)の数は年々増加しており、最近の自動車では20個~70個にもなる。これらの多くが相互に接続されており、一種の分散システムを構成している。自動車は、一つの機械・機器に例外的に多くのコンピュータが使われている例であるが、その理由として、自動車内の配線ケーブル(業界用語ではワイヤハーネスと呼ばれる)の重量の削減が大きな課題となっていることが挙げられる。

自動車にコンピュータを用いる目的としては、省エネルギーや低排気ガスに対する要求にこたえるために、より精密な制御が必要となっていることに加えて、安全性や利便性・快適性を上げることによって製品の付加価値を高めること、コンピュータによる制御でセンサなどのハードウェアを置き換えてコストダウンを図ることが挙げられる。

自動車に使われている数多くのコンピュータシステムを分類すると、たとえば次のように分類することができる。

- ▶ パワートレイン系(または駆動系)
- ▶ シャシー系(または走行系)
- ▶ ボディ系
- ▶ 情報・エンターテインメント系

これらのシステムごとに、システムに求められる要件は異なり、自動車内だけでも組み込みシステムの多様性を見ることができる。たとえば、走行系のシステム(例、ブレーキ制御システム)と、エンターテインメント系のシステム(例、カーオーディオ)では、信頼性要件が大きく異なることはいうまでもない。以下に挙げる特徴は、自動車内の多くの組み込みシステムに共通する特性である。

- 1) 製造個数が多く、部品のコストを下げることが重視される。そのため、リソース制約が厳しい。
- 2) 動作温度範囲が広く、ノイズの多い環境で動作しなければならないなど、環境条件が厳しい。このことも、厳しいリソース制約につながる。
- 3) 誤動作が人命にかかわる場合もあるため、極めて高い信頼性が求められる。
- 4) 高速に動作する機械を制御するため、高いリアルタイム性が求められる。

### 1.5.2 エンジン制御システム

自動車の中に使われる制御用のコンピュータユニットは、自動車業界では ECU(Electronic Control Unit)と呼ばれている。エンジン制御システムはパワートレイン系の中心となるシステムで、自動車を制御するためのシステムとしては、規模の大きいものである。メーカーにもよるが、最近では32ビットプロセッサが使われ始めてお

り、メモリやI/Oポートまで一つのチップに集積した1チップマイコンが使われていることが多い。また、ECU内には、1チップマイコンに加えて、センサやアクチュエータを接続するためのA-D変換回路などが実装されている。このコンピュータユニットが、エンジンルームや助手席の足元などに置かれている。

エンジン制御システムの働きは、スロットル開度やクランクの回転角、吸気温や排気中の酸素濃度などを計測する各種のセンサからのデータを元に、燃料の噴射量や点火タイミングなどを計算し、アクチュエータに出力することである。これらの処理を、各気筒に対して1サイクル(4サイクルエンジンでは2回転)ごとに繰り返して行う。

エンジン制御システムの特長としては、高速に回転するエンジンを精密に制御するために、短い応答時間方式にもよるが、ソフトウェアレベルで100 $\mu$ sオーダーが求められることと、排ガス規制や省エネルギーに対する要求をクリアするために制御が複雑化しており、ソフトウェアが大規模化していることが挙げられる。最近では、ソフトウェアの大規模化に伴って、ソフトウェアの開発や保守を容易にするために、リアルタイムOSが使われるようになってきている。

エンジン制御システムの場合、制御に失敗すると、エンジンが停止したり故障したりする場合もあるため、高い信頼性が求められるのはいうまでもない。また、燃料の噴射量や点火タイミングの計算は、それぞれのタイミングに間に合わなければ全く意味がなく、厳密なリアルタイム性が求められる。

ただし、計算が間に合わない場合に前回の計算結果を用いても、制御の精度は若干下がるが直ちにエンジンが停止するわけではない。そのため、すべての処理のリアルタイム性を厳格に守らなければならないわけではない。

### 1.5.3 自動車内のネットワークシステム

前に述べたとおり、自動車内のコンピュータの多くは相互に接続されて、一種の分散システムを構成している。従来は、この接続にはシリアル通信が使われることが多かったが、最近では車載ネットワーク(車載LAN)の適用が進みつつある。

車載ネットワークに求められる要件も、システムごとの特性によって異なる。たとえば、走行系のシステムを接続するネットワークは、高い信頼性と高速の応答性が求められるが、データ量は少ない。それに対して、情報系のネットワークは、データ量が多くなる。

そのため自動車内には、求められる要件に応じて、異なる規格に基づいた複数のネットワークが敷設される。また、複数のネットワーク間でデータを中継するゲートウェイが必要になる。たとえば、次のようなネットワークが敷設されることになる。

- ▶ボデー系のネットワーク
- ▶パワートレイン系・シャシー系のネットワーク
- ▶情報系のネットワーク

以上で見てきた自動車の事例は、組み込みシステムが応用に専用化して設計された結果、システム構築に多様な技術が使われる典型的な例となっている。車載ネットワークで見たように、1台の自動車の中にも複数の規格のネットワークが使われる。これは、パソコン用のLANが、Ethernetベースのものに収束しつつあるのと同様である。また、自動車内の各コンピュータシステムには、4ビットから32ビットまで多様なプロセッサが用いられるし、OSもシステムごとに最適なものが選定される(OSを使っていないシステムも多い)。

## 1.6 組み込みソフトウェア開発の最近の動向

.....

最近、組み込みシステムの制御対象となる機器が高機能化・複合化・デジタル化・ネットワーク化していることに対応して、システム自身も(ハードウェア的にもソフトウェア的にも)大規模化・複雑化している。多くの技術者が必要となっているにもかかわらず、教育体制の問題もあって、十分なスキルをもった組み込みシステム技術者の数は不足している。

それに加えて、time-to-marketの短縮(つまり、開発期間の短縮)や、開発コストの削減に対する要求も厳しい。これらの要因が、システムの品質や信頼性の低下につながっていることが危惧され、組み込みシステム開発は危

機能的な状況にあるということが出来る。

大規模・複雑化したシステムを短い期間で開発するためには、既存の設計資産やソフトウェアの再利用を進める以外に方法がないのが現状である。ここでいう再利用には、他社で開発された設計資産やソフトウェアの導入も含まれる。これを、組み込みソフトウェア開発のオープン化(または水平分業)と呼ぶ。また、既存のソフトウェア部品を組み合わせるシステムを設計する手法を、コンポーネントベース設計と呼ぶ。

従来の組み込みシステム開発においては、機器メーカーがすべてのソフトウェアを自社開発することで、機器の品質や信頼性を確保するという考え方がとられてきた。システムを構成するすべてのソフトウェアを自社開発することで、何らかの障害があった場合にも、障害の原因を確実に突き止め、それを取り除くことができる。

また、システムの振る舞いが予想できるため、予期できない要因により時間制約が満たせなくなるといった障害を最小限にできる。ところが、他社で開発されたソフトウェアの導入により、このような従来からの考え方が取れなくなる。この面からも、システムの品質や信頼性の確保が困難になっている。

大規模な組み込みシステムにおいては、OSを用いるのが以前より当然であったが、最近では個人向けの機器などコストダウン要求が厳しいシステムにまで、OSが利用されるようになってきた。その背景には、他社で開発されたソフトウェア部品(特に、ネットワーク接続のためのソフトウェア部品)をシステムに組み込む際に、既存のシステムとソフトウェア部品の共存のためにOSが必要となるケースが多い。

## 1.7 ITRON仕様と他のRTOS

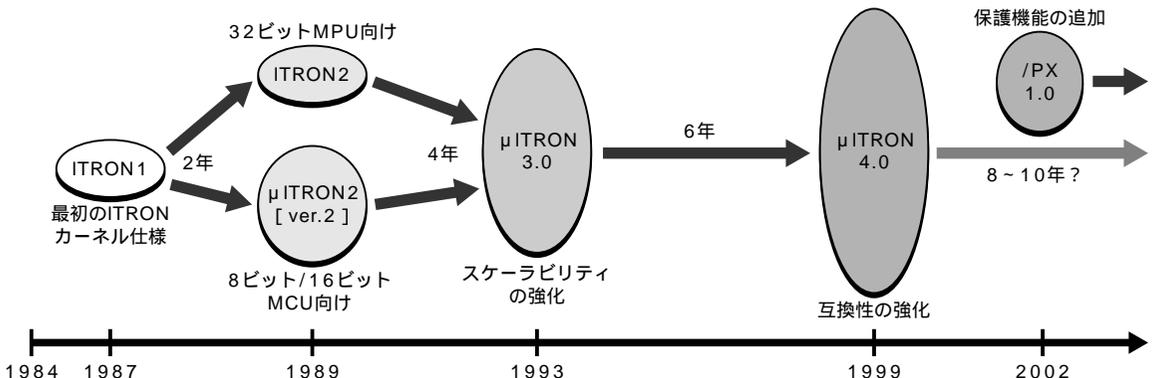
それぞれの機器メーカーが(場合によってはOSも含めて)すべてのソフトウェアを自社開発する従来の開発形態では、各メーカーが用いるOSを標準化する必要性は低く、図1.5にも示したように、多くの種類のOSが使われているのが現状である。

それに対して、既存のソフトウェア資産の再利用を進めるためには、ソフトウェア開発のベースとなるOSの標準化が不可欠となる。組み込みシステム向けのリアルタイムOSを標準化しようという試みはいくつか行われているが、その中で最も成功しているのがITRON仕様である。1.4節でも述べたとおり、国内で開発されている組み込みシステムの約4割にITRON仕様に準拠したOSが用いられているが、ソフトウェア資産の再利用の必要性から、この比率は大きくなる傾向にある。

ITRON仕様は、トロンプロジェクトの一環として標準化された、組み込みシステム向けのリアルタイムOSのAPI(Application Program Interface)の標準仕様である。1984年に検討が開始され、現時点までに4世代にわたるITRON仕様が標準化され、公表されている(図1.8)。

ITRON仕様は、多様な組み込みシステムに広く適用できるように、「弱い標準化」のコンセプトに基づいて標準化されている。「弱い標準化」とは、OSのもつ機能やその仕様を厳密に標準化するのではなく、アプリケーション

【図1.8】ITRON仕様の歴史



ョンからの要求やハードウェアの性質に合わせて最適化する余地を残すアプローチである。このアプローチにより ITRON 仕様は、8ビットから 64ビットの各規模のプロセッサに適用でき( 図 1.6 参照 )、組み込みシステムのあらゆる分野で広く使われている。

一方で、ITRON 仕様の OS 上で動作するように開発されたソフトウェアのポータビリティは犠牲になる面があるが、それよりもソフトウェア技術者の教育面を重視して標準化が行われてきた。

最新の  $\mu$  ITRON4.0 仕様では、ソフトウェア資産の再利用が重視されるようになってきたことに対応して、標準化の度合いを強くしたプロファイル規定が盛り込まれている。さらに最近では、ソフトウェアの大規模化・複雑化に対応するために、 $\mu$ ITRON4.0 仕様に対して保護機能を追加するための仕様(  $\mu$ ITRON4.0/PX 仕様 )も標準化されている。また、ITRON 仕様に関連して、ソフトウェア部品の API 仕様や、デバッグ環境とのインターフェース仕様の標準化も行われている。

組み込みシステム用の OS に関するもう一つの傾向として、汎用システム向けに開発された OS を、組み込みシステム分野の要求に合致するように改造して( または、そのまま )、組み込みシステムに適用しようという動きがある。典型的な例としては、Windows をベースとして組み込みシステム向けにリアルタイム性が強化された Windows CE や Windows Embedded と、Linux を組み込みシステム向けに改造した OS( 組み込み Linux と総称される )が挙げられる。