

# マイコン組み込み開発の現状と必要とされる人材

株式会社エンベデッド・システム 東京事業所 望月 亮治

## 1. はじめに

筆者はマイコン開発関連機器メーカーをはじめとして、約17年間「マイコン開発」にかかわるマーケティング部門での活動を行ってきた。

古くはハンドアSEMBルに始まったマイコンソフトの開発も、現在では高級言語、リアルタイムOS（以下RTOS）、TCP/IPでのネットワーク通信等、またハード面ではシステム・オン・チップ（以下SoC）などによるカスタムおよびセミカスタムCPUの出現などにより、劇的な変化を遂げてきた。特にこの3～4年の変化には目覚しいものがあり、筆者の顧客である電機、機械メーカーやソフト／ハード開発会社では、このような劇的な変化に対応するために多大な労力や費用をかけているのが現状である。

本発表では、このようなマイコン開発現場の状況と、求められる人材、また、どのような教育を行っていくことが求められているかをまとめる。

## 2. マイコンソフト開発現場の現状

### 2.1 プログラム容量の増加

まずは表1をご参照いただきたい。これは財団法人トロン協会が行ったアンケートの結果であるが、96～97年度とほぼ同率であったプログラムサイズ256KB未満と256KB以上の比率が変わり、99～00年度では約60%が256KB以上という回答になっている。また、97年度以前のデータがないために表中では紹介していな

表1 プログラムサイズ

単位：パーセント

メモリサイズ	96年度	97年度	98年度	99年度	00年度
64KB未満	25.4	24.7	25.7	13.5	17.1
256KB未満	23.2	25.3	21.7	25.7	24.8
1MB未満	27.7	25.2	25.2	31.6	28.0
4MB未満	23.7	24.7	17.8	16.1	20.3
4MB以上	—	—	9.6	13.1	9.9

表2 CPUの種類

単位：パーセント

CPU	96年度	97年度	98年度	99年度	00年度
4bit	2.1	1.2	0.8	0.4	0.2
8bit	17.1	16.0	14.1	7.7	10.2
16bit	34.5	32.2	34.6	26.9	25.1
32bit	45.2	48.4	46.6	58.1	57.6
64bit	1.1	1.2	1.5	3.8	2.2
専用(DSP等)		1.0	2.4	3.1	4.7

いが、16MB以上のプログラムサイズで開発を行ったという回答が、98年度3.6%、99年度3.1%、00年度5.4%あり、確実にプログラムの容量は増えているのうかがえる。

同様に使用したCPUの種類についての結果（表2）でも、96～98年度まで4～16ビットCPUと32ビット以上のCPUの比率はほぼ同率であったのが、99年度以降は約35対60に変化している。ただし、あくまでもこれは比率が顕著に出ている部分を比較したものであり、実際表中から読み取っていただきたいのは、「現在の組み込み開発の中心的なCPUは16～32ビットになっている」ということである。

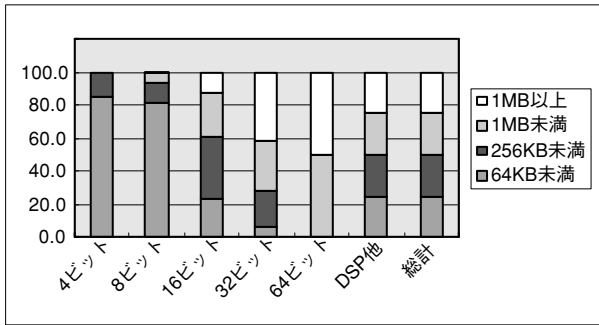


図1 CPUとプログラムサイズ

これは余談ではあるが、98年度といえばその年の後半からITRONにTCP/IPプロトコルスタックが搭載されるようになり、翌年3月頃よりITRON仕様のRTOSが爆発的に売れ始めたことが思い出される。

また、図1は97年度のデータに基づいてCPUの種類とプログラムサイズの関係についてまとめたものであるが、見てわかるとおり16ビットCPUではまだプログラムサイズ256KB未満が大半を占めるが、32ビット以上のCPUでは逆転して大半がサイズ256KB以上となっている。残念ながらこのデータは98年度以降集計されていないため、現在のデータがどのようなになっているかは紹介できないが、おそらく16～32ビットでのサイズ256KB以上の比率と32ビット以上のサイズ1MB以上の比率が上がっていると推測される。

## 2.2 開発環境の変化

組み込みにC言語が使用されるようになって久しいが、組み込み開発環境の現状を如実にあらわしているのが表3および図2である。調査は複数回答で行われたが、あえて単数で集計した。

97年度の時点で、すでに全体の中でC/C++が占める割合は73%に達しているが、その後の3年間で10%以上も伸び、約87%の開発者がC/C++を使用しているという回答となっている。この数値もやはり98年度から99年度へかけての伸びが顕著に現れている。

この傾向はRTOSの使用比率と相関関係があると考えられる。表4および図3はRTOSの使用状況に関するアンケートの集計結果であるが、96年度にほぼ同率であった「ITRON仕様OS（市販）+他市販OS」：「自社製OS+使用しない」の比率が、00年度には約

表3 使用開発言語

単位：パーセント

開発言語	97年度	98年度	99年度	00年度
アセンブラ	23.6	22.1	10.5	11.4
C	68.3	69.0	80.1	78.5
C++	4.7	6.3	7.3	8.4
Java	0.9	0.2	0.9	1.2
その他	2.5	2.4	1.2	0.5

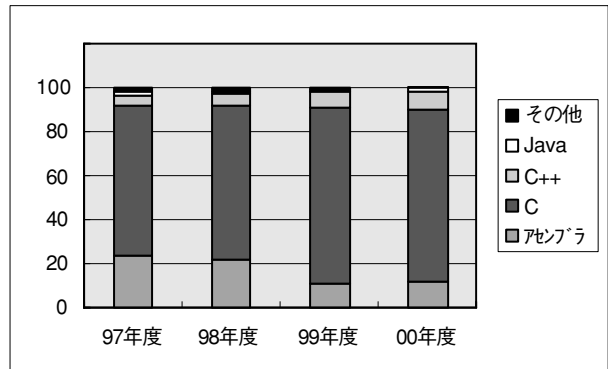


図2 使用開発言語

表4 RTOS使用状況

単位：パーセント

使用したRTOS	96年度	97年度	99年度	00年度
ITRON仕様OS	27.1	30.9	35.5	39.8
他市販OS	23.7	26.5	25.8	29.0
自社製OS	20.7	17.6	9.7	11.2
使用しない	28.5	25.0	29.0	20.0

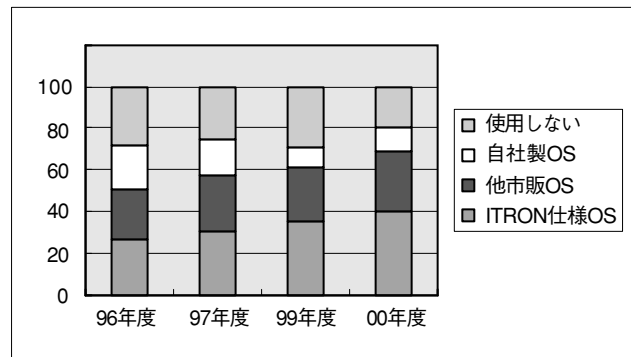


図3 RTOS使用状況

69：31と変化している。RTOSを使うメリットはいろいろあるが、なかでも「ソフトウェア部品の共有」、「開発コストの削減」は大きなメリットとなる。特にTCP/IPプロトコルスタックやPPP（モデム経由でのEthernet接続）などは、近年FAや計測器の分野にお

いて採用が著しいが、こういった機能を搭載するためにかかる費用（人件費、外注費）も期間も、市販のRTOSを購入して使ったほうが安上がりで開発期間も短く、さらに「メーカーの動作保証」によってリスク回避もできてしまうのである。

RTOSはC言語で書かれているのが一般的であり、メモリ等のリソースを必要とし、TCP/IP通信はCPUに高速動作を要求する。いわゆる98年後半から出始めたITRON仕様のTCP/IPプロトコルスタックを搭載したRTOSを購入したユーザは、必然的にCを使い、メモリの容量を増やし、大容量のデータ処理が可能な速いCPUを使い始めたと考えられる。

### 2.3 RTOS導入時の問題点

もう1つ興味深いデータがあるので紹介しておく。下記表5および図4はRTOS導入に当たっての問題点のなかで最大の比率である「技術者の不足」についてまとめたデータであるが、98年度を境として不足と感じる回答者の比率が増えている。RTOS全体では98年度から約5%の増加、ITRON系では99年度から同約5%の増加が認められる。これはRTOSを積極導入していかざるを得ない状況にあるが、技術者が不足しているために思うように進まないという企業が若干ではあるが増加していると考えられる。残念ながらデータ

表5 RTOS問題点（抜粋）

単位：パーセント

RTOS問題点	96年度	97年度	98年度	99年度	00年度
技術者不足	28.0	27.5	32.5	32.6	32.3
同ITRON	10.0	8.5	7.8	13.4	12.5

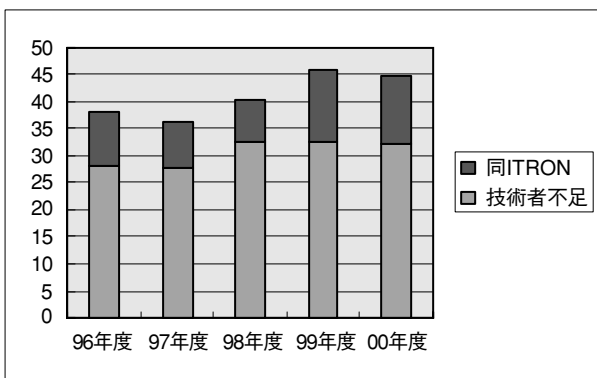


図4 RTOS問題点（抜粋）

が00年度までしか集計されていないため定かではないが、国内のさるRTOSメーカーの教育セミナーにおいては、3ヵ月分の日程をホームページで発表した後、ほぼ2週間以内にすべての予約が埋まってしまう現状を考えると、この技術者不足の傾向は、さらに強くなっていると予測される。

### 2.4 マイコンソフト開発に必要なスキル

現在のマイコンソフト開発において求められるスキルは、まず前述したC言語の普及率（約87%）、RTOS普及率（約80%）からわかるとおり、「C言語での開発経験」と「RTOSを使用した開発経験」の2つがあげられる。もちろんその前にプロセッサ自体の理論や動作といった基本を理解していることは必須であり、アセンブラでの開発経験は欠かせない。また、RTOSにおいてはさらに詳細な知識や経験が必要とされている。

以下表6にRTOSを理解するうえで要求される知識と経験をまとめる。

以上のように、RTOSを使ったマイコンソフト開発には、最低限C言語での開発経験が要求され、その経験のうえに段階を積み重ねて「エキスパート」と呼ばれるようになるのである。

一般企業の開発現場においては、通常段階4～5程度の開発工程が行われている。ここで重要となるポイントは「ドライバ開発」と、ドライバと自社アプリケーションの整合性をとる「カスタマイズ」である。い

表6 RTOSを使用した開発に要求されるスキル

段階	要求される知識・経験
段階0	C言語で組み込みのソフトウェアを組んだことがある
段階1	RTOSなしのプログラミングと、RTOSありのプログラミングの違いがわかる
段階2	タスクの概念を理解し、操作が可能
段階3	RTOS上でサンプルとして用意される各種周辺デバイスドライバ（プロトコルスタックを含む）の、自社ハードウェアへの移植およびカスタマイズが可能
段階4	RTOS上での、各種周辺デバイスドライバの独自開発が可能
段階5	各種通信用プロトコルスタックの開発が可能

かにドライバの移植と開発を行うか、またいかに移植あるいは開発したドライバを使ってアプリケーションを動作させ得るかがRTOSを使用する技術者に最も必要とされるスキルとなっている。

とはいえ短期大学校のように短い時間で多種類の技術の習得が要求されるケースでは、この部分にだけ時間を多く使うことは困難であり、基礎知識も不十分なままRTOSを理解しようとするのはかえって危険が伴う。余裕を持った習得計画が立てられるのであれば、ぜひ段階2までは理解することをお勧めしたい。

## 2.5 ハードウェアの変化

筆者がマイコンの業界へ入った17年前は、CPUといえばZ80か8085, 68XX等の8ビットマイコン, 8086, 68000の16ビットマイコン程度しか普及しておらず、やっと6809や8051などのワンチップマイコンが国内で出回り始めたばかりであったが、現在の国内のマイコン市場は正にワンチップマイコン時代。汎用CPUを探すほうが難しいほど多種類のワンチップマイコンが発売されている。そういった国内市場の中で近年SoCやSoPC (System on a Programable Chip) が注目を集めている。CPUコアのほかにあらかじめ用意されている周辺回路や自社独自設計の周辺回路などを組み合わせたASICマイコンであるが、SoCはASIC化されているのに対し、SoPCはCPLDやFPGAの上にCPUコアや周辺回路を載せてCPUを構築するタイプであり、ハードコア (コアの回路が固定化されて供給されるタイプ) とソフトコア (コアの回路がIPで提供され、設計者による自由なカスタマイズが可能なタイプ) に分かれている。

SoPCはSoCを作る前の試作などに使われるケースがほとんどであったが、以下のメリットによりSoPC自体も最終製品に搭載されるようになった。

- ・CPUと違い製造中止がない
- ・最終製品出荷後のハードウェア回路のカスタマイズやバージョンアップが可能
- ・ハードウェアの設計データが部品として流用可能

設計者は最終製品にSoPCを搭載し、ソフトコアであれば、必要ならばCPU内部を16ビットコアから32ビットコアに書き換えるなど、すべて書き換えてしまうことも可能である (もちろんこんなドラスティックな

表7 SoPC提供会社とサポート状況

メーカ	アルテラ	ザイリンクス
ハードコア	ARM9	PowerPC, tensilica
ソフトコア	Nios (16/32ビット)	MicroBlaze
対応OS	ITRON, Linux, 他OS	他OS

ことをする設計者はいないが)。

ハードウェア設計データの部品化に伴い、ソフトウェアの部品化も重要視されている。従来の機能ごとのソフト部品という概念ではなく、部品化された周辺回路ハードウェアに対応するドライバの部品化であり、これにはRTOSが威力を発揮する。ドライバをRTOS上の1つのタスクとできるように設計しておけば設計が変わってもほぼそのまま流用可能であるし、極端な話CPUが変わったとしても流用は可能である。ここでもRTOS上でのドライバ開発の必要性が認識されているのである。

最近是这样いったカスタムCPUの設計が簡単にできるツールも発売されており、またそういったCPUに対応したRTOSも発売されているので、高度な技能を習得することが要求される分野の教育においては、目先を変えてトライするよい機会なので、ぜひ経験してみることをお勧めする。(実売7~14万円程度)

## 2.6 設計およびデバッグ工程の重要性

マイコンシステム開発のプログラミング方法は、RTOSの普及により変わってきたことは先ほど述べたが、設計工程についても変化が起こったといえる。RTOSを使用することにより機能ごとのソフトウェア部品化が進み、プログラム全体が見通しやすくなってきた半面、ソフトウェア開発にかけられる期間が極端に短くなってきつつある。例えば携帯電話を例にとると「ソフト設計者150人体制で2ヵ月」という開発期間もまれではない。製品のライフサイクルが日を追うにつれ短くなってきている現状では致し方ないことであるが、「書いてみて、動かしてみ、直す」というような何度も手戻りする余裕はなくなってきた。「設計で仕様をカチッと決め、大人数で書きながら仕様どおりに動くか検証・デバッグし、動かす」が当たり前となっている。設計段階におけるバグつぶしを行うた



めには仕様書の作成が重要であり、完成度の高い仕様書が作成された結果、プログラミング工程の短縮化が図られるのである。またプログラミング段階においては、デバッグ工程を正確に行うことにより手戻りを少なくすることが可能となる。

仕様設計とデバッグ工程の重要性を正しく理解し実践することが、エキスパートへの近道といえるだろう。

### 3. どのような教育が必要か

#### 3.1 求められている人材像

以上、現在のマイコン組み込みシステム開発において重要と考えられている点を、主にソフトウェアの観点から述べてきたが、ここではどのような人材像が求められているかを、具体的に述べる。

- ・C言語でのマイコン開発ができる
- ・ワンチップマイコンでの開発経験がある
- ・RTOSと従来のプログラミング方法の違いを理解している
- ・仕様設計の重要性を理解している
- ・デバッグという概念に基づき、効率のよいプログラミングを行うことができる

さらに高度な技能を習得するのであれば、RTOSに関して以下の項目を追加したい。

- ・タスクの概念とシステムコールの動作を理解し、操作することができる

#### 3.2 どのような教育を行えばよいか

以上述べてきた事柄をもとに、現状のマイコン開発現場における人材に必要とされるスキルの習得を行う教育について、筆者なりに表8にまとめた。これらはすべて市販のソフトウェアやハードウェアを使って教育を行うことが可能であり、すでに導入されている施設もあるので、今後その効果に期待したい。

### 4. おわりに

マイコンのソフト/ハード開発の現場において、ここ数年で劇的な技術的変革が起こった。その結果として携帯電話やカーナビはネットワークコンピューティ

表8 教育内容

教育テーマ	使用CPU
コンピュータの基礎 ・構造 ・命令体系 ・割り込みとI/O ・周辺回路 ・アセンブラによる実習	8～32ビットワンチップ
C言語によるプログラミング ・割り込みとI/O制御 ・周辺回路制御 ・デバッグ ・実習	8～32ビットワンチップ
RTOS入門 ・RTOSの概念 ・従来のプログラミングとの違い ・タスクとセマフォ ・実習	8～32ビットワンチップ
RTOS応用 ・RTOSを使ったシステム設計手法 ・効率のよいプログラミング方法 ・ドライバの解説 ・実習(デバッグ実習を含む)	8～32ビットワンチップ
SoPC入門 ・SoPCの概要 ・CPU設計方法 ・CPU設計実習	ソフトコアCPU
SoPC応用 ・SoPCの設計 ・CPU設計実習(ファーム開発を含む) ・ソフトウェアプログラミング実習	ソフトコアCPU
システム設計概要 ・システム設計の概要 ・仕様書の重要性 ・使用作成実習	特定しない

ング端末となったが、そういった華やかな世界に一見隠されたように見えるFAや機械制御の世界にも、確実に技術変革の波が押し寄せている。

今回は、マイコン開発において必要とされる技術を紹介したが、今後はより効率のよい新規技術の教育用テキストやサンプルアプリケーションの提供を行い、また先生方が自由に参照およびダウンロード可能なインターネットサイトの構築をしていきたいと考えている。

#### 〈参考文献〉

- 1) 社団法人トロン協会編：『組込みシステムにおけるリアルタイムOSシステムの利用動向とITRON仕様OSに関するアンケート調査結果』（1996年度版～2000年度版）。