



C言語/OS/ICEを使って最先端の開発にチャレンジ

## 新世紀 マイコン教室

〈第1回〉マイコン開発ことはじめ

北野 優  
Masaru Kitano

世界初のマイクロプロセッサが出現して30年あまり、今日のマイクロプロセッサの発展にはすでに言を要しないところです。

巷ではユビキタス時代の到来というキャッチフレーズがもてはやされ、社会におけるマイコン技術の重要性はますます高まりつつあります。

そこで、本連載では、本誌2004年4月号付録マイコン基板を使用して、本格的なマイコン開発の考え方やその手法をわかりやすく解説していきます。

### 本連載の趣旨

#### ● 総合電子技術としてのマイコン技術

本稿では電子技術者を目指す方に向けてのマイコン開発の手法を解説します。

本稿の目指すところはソフト技術でも、マイコン技術でも、組み込み技術でもありません。

**総合電子技術としてのマイコン技術**です。

なぜ、総合電子技術としてのマイコン技術との問いへの答えは、昨今のグローバル化に伴う電子機器開発競争の市場変化にあります。

昨今、電子機器の開発現場では、比較的簡単に切り分けられる分野から順番にアウトソーシング化が進んでいます。最近では、海外の労働単価の安い国のソフト・ハウスでも高度な技術をもつようになって来ました。

今後は、論理回路設計さえも、値段の安い海外へ外注されることが多くなると思います。

つまり、ソフトウェアや論理回路などの単品技術では、よほどの高度な技術をもち合わせていない限り、先進工業国の賃金ではもはや成り立たなくなりつつあります。

幸い本誌が発行されているこの国には、先達の築いてくれた先進工業国としての豊富なリソースがあります。

つまり、素材から製品、システムにいたるまでの資源と知識がこの国では高度に集積されており、他国では到底望めないような情報と資材が豊富に利用できます。

本稿で言う総合電子技術とは、アナログ・デジタル、RF・AF・パワー・高精度、ソフト・ハードを問わず、さらに素材や工作技術、市場経営に至るまでの広範囲におよぶ知識とセンスを言います。

このような幅広い知識と技術はその習得に多大な努力と時間を要しますが、技術者としてその方向へ向かうことは決してまちがいではないと思います。

その中の組み込み型マイコンという、どこまでも奥深い分野の入り口を皆さんといっしょにのぞいてみようという試みです。

#### ● どんなことをするのか？

さて、上記のような大風呂敷を広げたからには、そのしまい方にたいへん苦勞するところですが、この際、潔く開き直って竜頭蛇尾のそしりはおおいに甘受し、各種入手できる便利ツールを駆使し「**いかに楽をして開発できるか**」をテーマに連載を進めます。

各社から提供、発売されている、または提供していただいた、C言語統合開発環境、インサーキット・エミュレータ、リアルタイムOSなど、最先端の開発ツールを使いながら「**最先端開発の最後尾**」に付くことを目標としています。

#### ● H8/300H TinyとC言語を中心とした連載

本稿では、主に使用するマイコンとして本誌2004年4月号に付録として収録され、すでにおもちの方も多**いH8/300H Tiny (H8/3694F)**を使用します。

2004年4月号を買って逃した方でも、互換品がサンハヤトから発売されているので安心です。

H8/300H TinyをはじめとするH8ファミリはたいへん洗練されたアーキテクチャで設計されており、C言語などの高級言語とも親和性の高いベストセラー・マイコンです。

性能もこの小さなマイコンで昔のスーパー・ミニコンに匹敵する演算能力を有しています。

ベストセラー&高性能ゆえ、開発ツールや技術情報も充実していますし、入手も簡単です。

また、本稿の**プログラム言語はすべてC言語を使用**します。マイコンの内部構造の理解が必要なアセンブラ言語は使いません。

さらに、本誌付録に特別に提供いただいたルネサス北日本セミコンダクタの**リアルタイムOS (RTOS) Smalight OS (体験版)**を使用します。

## 開発環境をそろえる ～マイコン三種の神器～

マイコン開発における「三種の神器」として以下の3種のツールを本稿では使用します。

### ■ C言語開発環境

#### ● 開発環境とは

開発環境とは、マイコン開発において使用するコンパイラ、エディタ、デバッガや書き込み用アダプタ、パソコンなど、マイコン開発に必要な装置、ソフトウェア一式のことを言います。

#### ● 統合開発環境とコンパイラ

本稿では、ルネサス テクノロジ純正の統合開発環境である**HEW3**とCコンパイラを使用します。

HEW3は、それ自身は内蔵されたエディタとCコンパイラやデバッガを管理するユーザ・インターフェース・ソフトウェアです。

CコンパイラやデバッガはHEW3に登録、連結して使用します。これらを総称して**ツールチェーン**といいます。ツールチェーンとして、HEW3に登録、連結されたCコンパイラやデバッガは以後HEW3上から操作できます。

つまり、いったんHEW3を起動するとプログラムの作成、コンパイル、デバックまで、すべてHEW3の画面から操作できるたいへん便利なソフトウェアです。

#### ● プログラムはC言語で書く

先に述べたように**プログラムはC言語で書きます**。理由はC言語で書けるからです。C言語は簡潔な表現で柔軟な記述ができる言語です。

C言語の欠点を挙げると数知れないですが、好き嫌いにかかわらずこれが業界の標準であり、本連載でもこれを使います。

C言語でプログラムが書けるということは、Cコンパイラが整備されているすべてのプロセッサのプログラムが記述できることにほかなりません。

連載の後半では、作成したプログラムをほかのプロセッサに移植することを試みたいと考えています。

C言語は論理回路の記述にも使われるようになってきており、今後は**電子技術者の必須習得言語**となるで



しょう。

#### ● アセンブラは使わない

マイクロプロセッサが直接処理できる機械語を表現した**アセンブラ言語は、本連載では使用しません**。

世の中にある各種のマイクロプロセッサは性能、機能を実現するために、それぞれ独自のアーキテクチャに基づいて設計されています。

アセンブラで記述するためにはそのアーキテクチャを完全に理解し、直接コントロール必要があります。

C言語ではそのほとんどをCコンパイラがやってくれますので、同じ目的を達成するにはるかに簡単に事を進めることができます。

ちょうど、運転免許を取得すればその免許の許す範囲内でエンジンやブレーキの形式がどうこうと気にすることなく、軽自動車でもスーパー・カー(死語)でも運転できるのと同じようなことです。

アセンブラで記述するほうが性能が良いという話がありますが、ある条件では確かに正解です。

ある条件とは、高度なプログラミング技術をもち、十分な時間と労力が投入できる場合でかつ、それにより得られる性能向上に価値がある場合です。たとえば、電池の消費を少しでも減らすためにCPUの動作時間を減らしたいというような場合です。

一般に、そのような時間を投入して性能を追及することはまれであり、ごく一部のソフト道を極めた人が行うべきことです。

私たち一般技術者は性能が不足するなら、プロセッサとCコンパイラが悪いと文句を言うのが正しい姿勢です。