

## SH-2/V850/ARM マイコンで制御するライン・トレース・カーで学ぶ ソフトウェア資産の再利用と 移植性の高いプログラミング方法

江崎 雅康, 堀 隆雄, 大竹 雅海

本章では、メーカーの異なる三つのCPUを使用してライン・トレース・カーを走らせる。同じ動作をさせるにもCPUが異なればプログラムも異なる。しかし、CPUによって変更が必要になる部分を理解すれば、どのCPUでも動作する汎用性のあるプログラムを作成できる。  
(編集部)

### 1. ソフトウェア開発の効率アップを目指して

#### ● 組み込み制御技術の今昔

マイコン制御の黎明期<sup>れいめい</sup>、8ビットCPUをアセンブリ言語によりプログラミングして開発されたシステムのハードウェアとソフトウェアの比率は50：50、もしくはハードウェアの方が大きかったかもしれません。そして、1人の技術者がハードウェアとソフトウェアの両方に責任を持ってシステム設計をすることも少なくありませんでした。

それから30余年が過ぎ、マイコン技術の進歩に伴い、ソフトウェアの比重が急速に大きくなってきました。製品開発の現場を見ると、ハードウェアとソフトウェアの比率は20：80、もしくはそれ以上になっていると思います。

そのような状況の中で、ソフトウェアの開発時間の短縮が重要になっています。そのため、ソフトウェア資産の再利用や移植性、そして仕様記述言語はこのような背景の中で必要とされている技術です。

#### ● プログラムの移植性とは

組み込みシステムの場合、ハードウェアA用に開発したソフトウェアAが、異なる仕様のハードウェアBでそのまま動作することはほとんどありません。ソフトウェアAを元にハードウェアBの仕様に合わせてソフトウェアを変更します。この作業を「移植」と呼びます。

ハードウェアAとハードウェアBの違いがどの程度あるかにもよりますが、いくつかの変数と定義ファイルを書き換えるだけで移植が完了する場合は「移植性のよいプロ

グラム」といえます。しかし、ソフトウェアの書き方によっては、ソース・プログラムを最初から書き直さざるを得ない場合もあります。これを「移植性の悪いプログラム」と呼びます。

#### ● 移植性を妨げる要因

CPUの違いは、移植性を妨げる大きな要因です。ビット数やメモリ容量が同じでも、メーカーによって命令セットや命令コードが全く異なります。マシン語やアセンブリ言語で記述したソース・プログラムは移植できません。しかし、C言語で記述された制御アルゴリズムの移植は可能です。移植は可能ですが、CPUによって異なる部分を変更可能なようにプログラムされていないと簡単に移植はできません。移植性のよいプログラムとは、CPUに依存する部分と依存しない部分をきちんと分けて記述してあることです。

また、周辺デバイスの違いも移植性を阻む大きな要因です。CPUに内蔵されている

- GPIO
- シリアル通信コントローラ
- カウンタ/タイマ
- A-Dコンバータ
- 割り込みコントローラ

といった周辺デバイスの基本機能は同じですが、アーキテクチャの制御仕様や性能はチップごとに異なります。

開発環境の違いも移植性を阻害する要因です。各チップ・メーカーの開発ツールは、伝統的なC言語の処理系を超えて便利な関数やライブラリを用意しています。C言語自体の移植性が非常に高かったとしても、開発環境固有の関