

ITRON サブプロジェクト

— 第2フェーズへの展開 —

□ 豊橋技術科学大学 高田 広章
□ (株) 東芝 田丸 喜一郎

はじめに

ITRON サブプロジェクトは、組み込みシステム用のリアルタイムオペレーティングシステム（以下、RTOSと略記する）と、それに関連する仕様の標準化を行うプロジェクトである。1984年にプロジェクトを開始してから現在までの約15年の間に、三世代のITRONリアルタイムカーネル仕様が策定・公開してきた^{★1}。また、その普及活動にも力を入れてきた結果、多くの種類のプロセッサ用実装され、きわめて多くの組み込み機器に利用されている。

図-1のグラフは、(社) トロン協会が1997年末から1998年始めにかけて国内で行ったアンケート調査による、組み込み機器に利用されているRTOSのシェアを示している。この調査結果では、ITRON仕様のRTOSを用いている機器が全体の約28%、RTOSを用いている機器の中では約38%を占めており、ITRONリアルタイムカーネル仕様が国内ではデファクト標準として定着していることが分かる。応用分野別では、AV機器、家電製品、個人用情報機器、通信端末など、1人ないしは一家が1台持つ機器において、ITRON仕様のRTOSが多く使われている。

ITRON サブプロジェクトは、この成果をベースとして、1996年頃から第2フェーズへの展開を図っている。第2フェーズにおいては、第1フェーズにおけるリアルタイムカーネル仕様に焦点を置いた標準化活動から、ソフトウェア部品などの周辺仕様へと範囲を拡大して標準化活動に取り組んでいる。

本稿では、最初に、ITRON サブプロジェクトの第1フェーズにおける活動とその成果を概観し、一定の成功を収めることができた要因について議論する。次に、現在進行中の第2フェーズの標準化活動について、その狙いと現時点までの成果を中心に紹介する。最後に、ITRON仕様の海外においても普及させるための活動に

★1 マルチタスクサポートを主な機能とするRTOSの中心となるモジュールを、リアルタイムカーネルないしはリアルタイムモニタと呼ぶ。μITRON仕様には、I/Oを扱う機能は含まれておらず、リアルタイムカーネル仕様と呼ぶのが妥当である。それに対してμITRON仕様に準拠した製品は、I/Oを扱う機能を持っている場合もあるため、本稿ではITRON仕様のRTOSと呼ぶことにする。

ついて紹介し、国際的な標準とするための条件について議論する。

第1フェーズの活動と成果

ITRON サブプロジェクトの第1フェーズにおいて策定された仕様とその公開時期を図-2に示す。以下では、それぞれの仕様の策定の背景と利用状況について概観する。仕様の技術的な内容や実装技術については、文献1)で紹介しているため、ここでは省略する。

プロジェクト開始からITRON1まで

ITRON サブプロジェクトは、トロンプロジェクトの最初のサブプロジェクトとして1984年に開始された。当時すでに、米国の半導体メーカーやソフトウェアメーカ

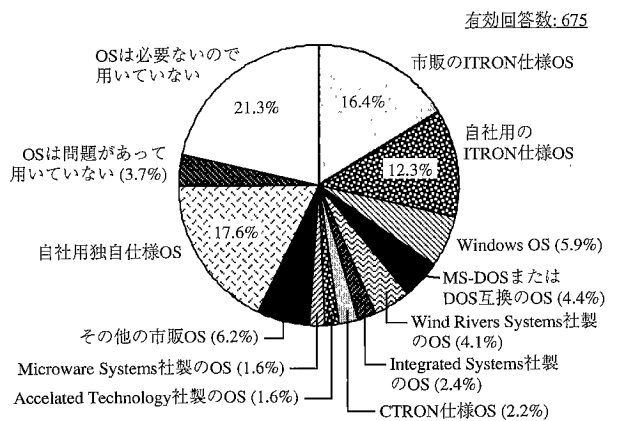


図-1 組み込みシステムに使用したOS

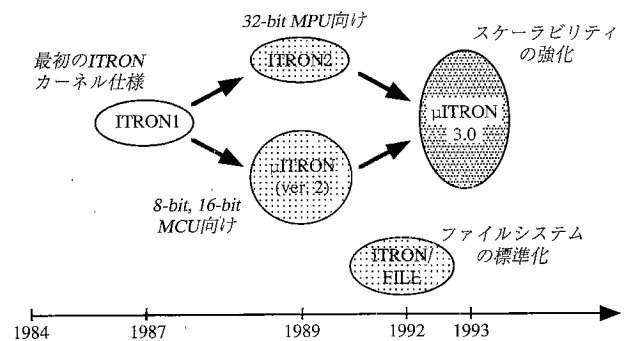


図-2 第1フェーズのITRON仕様

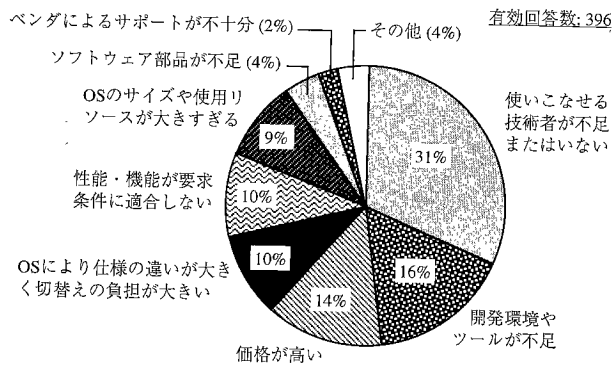


図-3 RTOSの問題点

によりいくつかのRTOSが開発されており、日本の半導体メーカーも、自社のマイクロプロセッサの適用範囲を広げるために、独自にRTOSを持つ必要性が高まっていた。1993年には、日本電子工業振興協会（JEIDA）のマイクロコンピュータ技術委員会において、RTOSに対する要求仕様がまとめられている。この要求に答える標準的なRTOS仕様を開発することを目的に開始されたのが、ITRONサブプロジェクトである。

ITRON仕様の設計にあたっては、当時のマイクロプロセッサが計算能力・メモリ容量ともきわめて小さかったことから、プロセッサの仮想化を行わず、高速な応答性を可能とすることが重視された。また、技術者の教育を容易にすることを重視し、用語やシステムコール名の一貫性に注意が払われた。逆にソフトウェアの移植性については、ある程度犠牲になってもやむを得ないと考えた。これらの方針は弱い標準化と呼ばれ、現在に至るまでITRON仕様の基本的な設計方針となっている。前述のアンケート調査でRTOSの問題点について質問した結果（図-3）からは、「技術者の不足」が現在においてもRTOS利用上の最大の問題点であり、また多くの技術者が「使用リソースが大きすぎる」「性能・機能が要求に合致しない」という性能上の問題を指摘しており、15年を経過しても本質的な問題が変わっていないことが分かる。

1984年当時は、8086や68000などの汎用16ビットマイクロプロセッサが主流であった。プロジェクトに参加した半導体メーカー各社はそれぞれ、仕様の検討と並行して、これらのプロセッサ上にITRON仕様のRTOSの開発を進め、そこから出てきた要求事項や経験を仕様フィードバックした。その結果、1987年には最初のITRONリアルタイムカーネル仕様であるITRON1仕様が開発された。ITRON1仕様は、RTOSに関する基礎的な技術と経験を得る上で意義が大きかった。

μITRONとITRON2

半導体メーカーが、組み込み用途に特化した8ビット、16ビットのワンチップマイコンに力を入れるようになると、それらを用いた小規模な組み込みシステムに適用するために、より小さなメモリ容量で実現可能なRTOSが求められるようになった。この要求に答えるために、

μITRON仕様（Ver 2.0）が検討され、1989年に公開された。μITRON仕様では、ITRON1仕様の機能を整理・縮小するとともに、どの機能を実装するかを実装者の自由とし、応用ごとに必要な機能だけ実装することを可能としている。

このμITRON仕様は、ITRONサブプロジェクトの中心メンバーである半導体メーカー各社のニーズに合致したことから、数多くの8ビット、16ビットプロセッサ上に実装された。実際、多くの半導体メーカーは、組み込みシステム向けに自社独自のプロセッサを開発すると、アセンブラ、Cコンパイラと並んでμITRON仕様のRTOSを開発する体制となっており、トロン協会に登録されているものだけでも、40を超えるμITRON仕様準拠のRTOSが開発されている（次に紹介するμITRON3.0仕様準拠のものも含む）。

このように多くのμITRON仕様のRTOSが開発されたのは、いうまでもなく、μITRON仕様のRTOSの採用がユーザ（組み込み機器メーカー）にメリットがあり、その結果きわめて多くの組み込み機器に採用されたためである。μITRON仕様のユーザに対するメリットとしては、限られた計算資源しか持たないワンチップマイコン上でも実用的な性能を発揮できたことに加えて、この規模のRTOSとして国内外に競合製品がなかったこと、プロセッサを多数購入するユーザに対して半導体メーカーがRTOSからアプリケーション開発まで含めたトータルなサポートを提供したことなどがあげられる。

これらのメリットに加えて、ほとんどの組み込みシステム向けプロセッサにμITRON仕様のRTOSが実装された結果、μITRON仕様を使用していれば、次期の製品でプロセッサを変更することが容易になるという標準化本来のメリットも生まれることとなった。また、参考文献やセミナーが豊富に提供されていることも、重要な標準化メリットである。ITRONサブプロジェクトとしても、μITRON仕様の普及のためのセミナーの開催や、標準的な参考書²⁾を用意するなどの活動を行っている。さらに、ユーザ企業の中には、μITRON仕様を社内標準と定めるところも出てきた。これは、社内の部門ごとに別々のRTOSを利用すると、技術者が部門を移動するたびに再教育が必要になるためである。

また、組み込み機器メーカーが自社でRTOSを開発する場合にも、μITRON仕様準拠した仕様とすることが多くなってきた。これには、上で述べた標準化メリットに加えて、μITRON仕様完全にオープンな仕様であること（仕様書は一般書籍として販売されており、それに基づいたRTOSを自由に実装できる）が前提となっていることはいうまでもない。

一方、μITRON仕様と並行して、より高性能な32ビットマイクロプロセッサを対象として、ITRON1仕様よりも機能を充実させたITRON2仕様が策定され、

μITRON仕様と同じ1989年に公開された。ITRON2仕様は、TRON仕様チップ用に複数のメーカーによって実装されたが、TRON仕様チップの応用が広がらなかったために、ITRON2仕様のRTOSの応用事例も限定されたものとなった。他のハイエンドマイクロプロセッサ用のRTOS仕様としてITRON2仕様を活用する道もあったが、多くの競合製品（ほとんどは米国製）があったため、普及は難しかった。また、当時はまだハイエンドマイクロプロセッサを必要とする組み込み機器が数量的に少なく、半導体のビジネスとしては大きい市場にはなっていないことも背景にある。

μITRON3.0

1990年代に入り、半導体メーカー各社が組み込みシステム向けの32ビットプロセッサの開発を始めると、元来小規模なプロセッサ向けに設計されたμITRON仕様が32ビットプロセッサにも実装されはじめ、μITRON仕様の機能不足が問題になってきた。そこで、μITRON仕様をベースにITRON2仕様の一部の機能を取り込み、8ビットから32ビットまでの広い範囲のプロセッサをカバーできるスケラブルな仕様とすることを旨としてμITRON3.0仕様が設計され、1993年に公開された。

μITRON3.0仕様における新しい機能として、1つの組み込み機器内に閉じたネットワーク（たとえば、コピー機には複数のプロセッサが使われており、連携動作しているものがある）をサポートするための接続機能が導入された。

μITRON3.0仕様の公開後も、古いμITRON仕様に基づいた製品が多く残っており、移行の速度はあまり早くなかった。これは、バブル経済崩壊の影響で製品の新規開発費が圧縮されたことも考えられるが、古い仕様でも大きな問題がなかったという理由が大きい。一度古い仕様に基づいて開発された製品を、μITRON3.0仕様準拠にバージョンアップすることは開発コストの問題から難しく、新規に開発される製品から新しい仕様に準拠させているメーカーが多い。現時点でも、古いμITRON仕様準拠のものの方が、登録製品の数では多くなっている（新しい製品で未登録のものが多いことも原因の1つである）。

プロジェクト開始から約10年が経過した1990年代半ばになると、組み込みシステムの複雑化に伴ってRTOSを用いるシステムが増加してきたこともあり、μITRON仕様を利用する機器がさらに増えてきた。また、半導体メーカーとは独立のソフトウェアメーカー（サードパーティ）がμITRON仕様のRTOSを開発するケースも増えてき

た。従来からRTOSを販売してきたソフトウェアメーカーにとっても、μITRON仕様に準拠したRTOS製品の方がユーザに受け入れられやすいという明らかなメリットがある。さらに、プロジェクトに参加していない半導体メーカーからμITRON仕様のRTOSの開発・販売を委託されるケースや、自社用に開発したμITRON仕様のRTOSを外販するケースもある。特に前者のケースでは、海外の半導体メーカーから日本のソフトウェアメーカーが委託を受けるといった興味深い例も複数見られる。

また、μITRON仕様がデファクト標準となるにつれて、コンパイラ、デバッガ、ICEなどの組み込みシステム用のソフトウェア開発環境のメーカーや、通信プロトコルスタックなどのソフトウェア部品メーカー、CASEツールのメーカーも、μITRON仕様をサポートしはじめた（これらの中には海外のメーカーが多い）。実際、最近になって、これらのソフトウェアメーカーを中心に、ITRONサブプロジェクトへ参加する企業が急増している。

最近では、海外のソフトウェアメーカーがμITRON仕様準拠したRTOSを開発するケースも増えてきている。一例として、Cygnus Solutions社によるフリーオープンソースのRTOSであるeCosがあげられる³⁾。これには、μITRON仕様をサポートすることが日本市場へ入り込む近道であること、他に同じ規模のRTOSの標準仕様がないこと^{☆2}が大きな理由となっている。

成功の要因

以上で、μITRON仕様がデファクト標準として定着してきた経緯を紹介したが、その成功の要因を今一度整理してみる。

最初に、成功の大前提ともいえるのが、産業界のニーズを的確に把握し、それに答える標準仕様を提示してきた点である。これを裏付けるデータとして、図-4に前述のアンケート調査でITRON仕様の長所について質問した結果を示す。このグラフから、ITRON仕様の策定にあたって重視した項目が上位にあがっており、また図-3の問題点ともほぼ符合していることが分かる。それに対

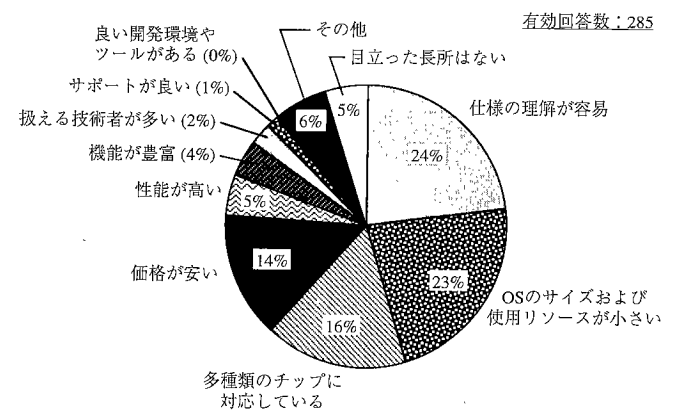


図-4 ITRON仕様OSの長所

☆2 RTOSの標準仕様としてはリアルタイムPOSIXが有名であるが、リアルタイムPOSIXはμITRONと比べて規模的にはかなり大きいOS仕様である。最近、自動車制御応用に焦点を当てたRTOS仕様としてOSEK仕様が標準化されたが、これはμITRON仕様とほぼ同じ規模のRTOS仕様である。

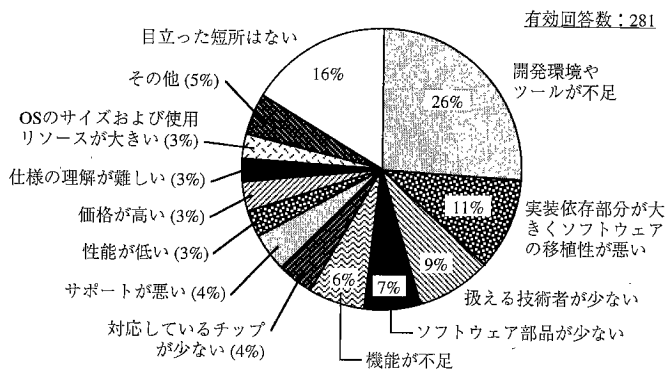


図-5 ITRON仕様OSの短所

第2フェーズの標準化活動

ITRONサブプロジェクトの第2フェーズにおいては、従来のリアルタイムカーネル仕様へに焦点を置いた標準化活動から、以下で紹介するような周辺仕様へと標準化活動の範囲を拡大して取り組んでいる。

前述したように、μITRON仕様はきわめてコンパクトに実現できるリアルタイムカーネル仕様である。一方、組込みシステムの大規模化・複雑化は急速に進んでおり、カーネル上で動作するソフトウェアのすべてを組込み機器メーカーだけで構築することは難しくなっている。そのため、カーネル仕様だけの標準化では不十分になってきており、カーネル上で動作するソフトウェア部品（ミドルウェア）のための標準化が重要性を増している。ソフトウェア部品のための標準化としては、ソフトウェア部品そのもののインタフェースの標準化と、ソフトウェア部品が流通する前提条件を整えるための標準化の両方が必要と考えている。

標準化範囲拡大の他の方向性として、ソフトウェア開発環境関連の標準化と、応用分野ごとの要求に対応した標準化があげられる。前者については、開発環境面が弱いことがITRON仕様のRTOSの問題点であるという指摘が多く（図-5）、開発環境面を充実させるための活動が重要となっている。

以下では、ITRONサブプロジェクトの第2フェーズにおいて取り組んでいる標準化活動について、その狙いと現時点までの成果を中心に紹介する。

ソフトウェア部品のインタフェース標準化

組込みシステムの構築にソフトウェア部品を使うことが多くなると、ソフトウェア部品のインタフェースの標準化が重要となるのはいうまでもない。ただし、組込みシステム用にも、多くの分野のソフトウェア部品があり、そのインタフェースの標準化にあたっては、重要性が高い分野から着手すべきと考えられる。

現時点までにITRONサブプロジェクトでは、TCP/IPプロトコルスタックとJava実行環境の2つの分野において、ソフトウェア部品インタフェースの標準化を行った。このうち、Java実行環境とのインタフェース標準化については、本特集のJTRON仕様に関する記事で扱うのでここでは省略する。

最近、組込みシステムの分野においても、インターネットとの接続が重要性を増している。現在、TCP/IPプロトコルスタックのAPI（Application Program Interface）として最も広く使われているのはソケットインタフェース（またはそこから派生したインタフェース）であるが、ソケットインタフェースは特に小規模な組込みシステムには不十分な点があり、それに代わるインタフェース仕様が求められていた⁵⁾。実際、μITRON

して前述の接続機能については、必ずしも要求に答える仕様となっていなかったことが、広く受け入れられなかった原因であったと考えている。

一方で、技術的な優劣で標準が定まるわけではないことは広く知られており、μITRON仕様が普及した背景にも、技術面以外の大きい要因がある。

前述したように、μITRON仕様には有力な競合相手がなかった。とりわけ、ソフトウェア産業で優位に立つ米国に有力な競合製品がなかった。さらに、競合製品がなかった理由を考えると、日米の組込み機器産業の違いに言及する必要がある。すなわち、米国における組込みシステム技術は、軍事産業や航空宇宙産業などの大規模なシステムに引っ張られて発展してきた。それに対して日本においては、AV機器や家電製品など、システムとしては小規模だが大量生産される分野が有力である。特に、ITRONサブプロジェクトの中心メンバーである半導体メーカーにとっては、大量生産品が大きなビジネス対象であり、μITRON仕様はそのような分野を主たるターゲットとして策定された。

少量生産品では開発コストの削減が重要であるのに対して、大量生産品においては、最終製品のコストを下げることが重視される⁴⁾。そのため大量生産品では、システムに必要なハードウェア資源の削減に対する要求が厳しく、RTOSにも実行時オーバヘッドや使用メモリ量が少ないことが求められる。μITRON仕様のRTOSは、仕様面でも実装面でもこのような要求に答えることが重視されているのに対して、(相対的に)大規模なシステムをターゲットとする米国のRTOS製品はそのような要求に答えることができなかった。逆に、ITRON2仕様がμITRON仕様と比較して思うように普及しなかったのは、少量生産品の分野で重要となるソフトウェア開発環境の面で弱かったことも原因になっていると考えられる。

まとめると、日本が優位性を持つ産業分野には小規模な組込みシステムにかかわるものが多く、μITRON仕様のような小規模なシステム向けのRTOSに対するニーズが日本において最も大きかったことが、日本発のμITRON仕様がデファクト標準となり得た大きな理由であるといえる。

仕様のRTOS上で動作するTCP/IPプロトコルスタック製品について調べたところ、その多くがソケットインタフェースに加えて独自のAPIを持っていることが分かった。ITRON TCP/IP API仕様は、ソケットインタフェースの問題点を解決し、コンパクトで効率のよいTCP/IPプロトコルスタックの実現を可能にするためのAPI仕様として設計されたもので、1998年5月に公開された。現在、複数の企業が、この仕様に準拠したTCP/IPプロトコルスタックを開発中である。

これらのソフトウェア部品のインタフェース仕様は、概念や用語をITRONリアルタイムカーネル仕様と整合させ、μITRON仕様のRTOSを使っている人には、容易に学習できるよう配慮している。それと同時に、他のRTOS上でも利用可能なインタフェースとなるよう、できる限りμITRON仕様のRTOSの機能に依存しないような仕様としている。

μITRON4.0リアルタイムカーネル仕様

μITRON仕様のRTOS上で動作するソフトウェア部品が流通する前提として、μITRON上のソフトウェアの移植性が高いことが求められる。また、システムの大規模化からも移植性の向上が求められているが、従来のμITRON仕様においては、弱い標準化の方針により、同じ仕様に準拠していても製品により仕様が異なる場合が多い。実際、図-5においても、ソフトウェアの移植性が悪いことがμITRON仕様の短所としている技術者が多い。一方で、弱い標準化の方針により、広い範囲のプロセッサへ適用できるスケラビリティが実現されており、μITRON仕様の最も重要な特長の1つとなっている。

弱い標準化の利点を維持しつつ、ソフトウェアの移植性を向上させるために、弱い標準であるμITRON仕様の中に標準的な機能集合（これをスタンダードプロファイルと呼ぶ）を厳密に定める方針が考えられ、この方針に従って現在標準化検討が進められているのがμITRON4.0仕様である。μITRON4.0仕様のRTOS上で動作するソフトウェア部品は、スタンダードプロファイルに含まれる機能のみを用いて構築することが推奨される。一方、ソフトウェア部品の利用が必要な応用分野を狙ったRTOS製品は、スタンダードプロファイルの機能を満たすことが推奨される。

それに加えて、μITRON4.0仕様においては、ハードリアルタイムシステムの構築をサポートする機能追加と、自動車制御分野における要求事項の取込みを行っている。後者については、この分野においてμITRON仕様のRTOSが使われている率が低いことから、分野独自の要求事項を洗い出す活動を行ったが、その成果を反映するものである。

第2フェーズの活動において、カーネルから周辺仕様へ重点を移そうとした結果、再びカーネル仕様を検討す

ることとなったわけだが、カーネル仕様のバージョンアップには、μITRON3.0仕様を出してから5年間の半導体技術の発展に対応するためという意味もある。実際、5年前にはオーバーヘッドが大きいため標準化しないと判断した機能で、μITRON4.0仕様のスタンダードプロファイルに含まれるものがある。

現在、μITRON4.0仕様の標準化検討は最終局面まできており、近日中に仕様を公開できる予定である。

その他の活動

上で紹介した以外では、最近、μITRON仕様のRTOSとデバッグ環境との間のインタフェースを標準化する活動を開始した。前述したように、μITRON仕様のRTOSは開発環境面が弱いといわれているが、これは、RTOSを開発しているメーカーが、自社で開発環境を開発していないか、開発していたとしても得意ではないケースが多いためである。そのため、専業メーカー製の開発環境が使われることが多いが、μITRON仕様のRTOSと開発環境（特にデバッグ環境）との間のインタフェースが標準化されていないために、開発環境メーカーがそれぞれのμITRON仕様のRTOSに個別に対応せねばならず、その対応が進んでいないのが問題と考えられる。そこで、μITRON仕様のRTOSとデバッグ環境との間のインタフェースを標準化することで、デバッグ環境を仕様に準拠した各RTOSに対応させることが容易になり、優れた開発環境を使うことが容易になると考えられる。

また、難しいといわれるデバイスドライバの設計を容易にするためのデバイスドライバ設計ガイドラインや、リアルタイム制約を持ったアプリケーションとソフトウェア部品の共存を図るためのアプリケーション設計ガイドライン⁶⁾の検討作業も行っている。

その他に、ITRON仕様のRTOS上で動作するソフトウェア部品群を実装して、フリーソフトウェアとして配布するための開発プロジェクトを進めている。このプロジェクトは、情報処理振興事業協会（IPA）が公募した「次世代デジタル応用基盤技術開発事業」の採択テーマの1つとして推進されているものである。

これらの標準化活動を通じて、μITRON仕様の短所を克服するとともに、常に変化する要求に対応していくことを目指している。

国際標準へのアプローチ

ここでは、ITRONサブプロジェクトで行っている海外への普及活動を概観するとともに、国際的なデファクト標準へ向けての展望について述べる。

ITRONサブプロジェクトでは、μITRON仕様が国内においてデファクト標準として定着してきた1995年頃

から、海外への普及活動を本格的に開始した。具体的には、組込みシステム分野において世界最大の展示会である Embedded Systems Conference への出展や、それと同時期にセミナーを開催するなどの活動を行っている。また、IEEE Computer Society Press から μ ITRON3.0 仕様に関する書籍を発行した⁷⁾。アジア地域においても、中国で ITRON に関するセミナーを開催するなどの普及活動を行っている。

現在の μ ITRON 仕様の海外における状況とその背景は次の通りである。

(1) 前述の通り、海外の半導体メーカー、RTOS メーカー、ソフトウェア開発環境メーカー、ソフトウェア部品メーカー、CASE ツールメーカーで、 μ ITRON 仕様をサポートしている企業が増えている。この理由として、これらのメーカーが日本でビジネスをするために、 μ ITRON 仕様をサポートすることが必要になっていることが大きい。さらにこの背景として、RTOS や開発環境などの組込みシステム開発用ツールの分野においては、日本は全世界の市場の 3 割程度を占めていると推定され、これらのメーカーにとって日本市場は無視できないという事情がある。とりわけ、日本には組込み機器メーカーとして世界的に有力な企業が多く、それらの企業が μ ITRON 仕様のサポートを要求することで、 μ ITRON 仕様のサポートを始めるケースが多い。

(2) 日本の組込み機器メーカーからソフトウェア開発を受注している海外の企業において、発注元の要求により μ ITRON 仕様の RTOS を使ってシステム開発を行うケースがある。機器メーカーの海外子会社においても同様のケースが見られる。

(3) 海外の組込み機器メーカーが、日本の半導体メーカーのプロセッサを採用した場合に、プロセッサと一緒に μ ITRON 仕様の RTOS を採用するケースがある。

(1) の状況から、組込みシステム開発用ツールのメーカーに対しては μ ITRON 仕様の周知度がかなり高いのに対して、(2) や (3) のような事例はそれほど多くはなく、海外の組込み機器メーカーに対する μ ITRON 仕様の知名度はあまり高くないのが現状である。以下では、このような現状から、 μ ITRON 仕様を国際的なデファクト標準とするための展望について議論する。

前述したように、米国においては小規模な組込みシステムは重視されてこなかったが、最近では、軍事費削減が大きな契機となって、デジタル情報家電など、 μ ITRON 仕様が得意としている民需品を重視する傾向にある。そのため、米国においても μ ITRON のような規模の RTOS が必要とされるようになってきた。一方で、競合製品も同じターゲットを狙っており、これまでの技術開発の積み重ねや日本での普及という実績をベースに、競合製品も μ ITRON 仕様をサポートする（その時点で、競合製品ではなくなる）という状況にしていく

ことが必要である。

また、半導体メーカーや開発用ツールのメーカーによる μ ITRON 仕様のサポートは着実に進展しており、それらが日本市場だけでなく海外市場においても積極的に販売されるようになり、英語での参考文献やセミナーが豊富に用意されるようになってくると、 μ ITRON 仕様の標準化メリットにより、海外においても急速に普及する可能性は十二分にある。さらに、現在進めているように、 μ ITRON 仕様の RTOS 用のソフトウェア部品や開発環境を充実させる方向性の標準化活動を通じて、標準化メリットを増していくことが重要と考えられる。

そのためには、海外における ITRON サブプロジェクトの支援者を増やすことが必要で、従来行ってきた普及・広報活動に加えて、標準化の検討を英語ベースで行い、海外の企業の参加を促すことが重要であると考えている。

おわりに

本稿では、ITRON サブプロジェクトの現状と μ ITRON 仕様が普及した理由、それを受けての第 2 フェーズの標準化活動とその背景について述べた。 μ ITRON 仕様は、言うなれば日本の得意分野のための OS 仕様であり、この分野で国際標準を日本から発信できなければ、他の分野ではより難しいものと思われる。

コンピュータ産業の過去の歴史からも、OS やプログラミング言語の普及には長い時間がかかることは明らかである。 μ ITRON 仕様が国内でデファクト標準と呼べるようになるまでにも、プロジェクト開始から 10 年を超える時間がかかっている。 μ ITRON 仕様が国際標準となる条件を備えていたとしても、国際的な普及にはまだ数年の時間が必要と考えられ、継続的な努力が求められる。

最後に、ITRON サブプロジェクトに関するより詳しい情報については、ITRON ホームページ⁸⁾を参照いただければ幸いである。

参考文献

- 1) 高田広章, 田丸喜一郎他: ITRON サブプロジェクトの現状と展望, 情報処理, Vol.35, No.10, pp.903-909 (Oct. 1994).
- 2) 坂村 健編: ITRON 標準ガイドブック 2, パーソナルメディア (1994).
- 3) <http://sourceware.cygnus.com/ecos/>
- 4) 中本幸一, 高田広章, 田丸喜一郎: 組込みシステム技術の現状と動向, 情報処理, Vol.38, No.10, pp.871-878 (Oct. 1997).
- 5) 高田広章: 組込みシステムに適した TCP/IP API, コンピュータシステム・シンポジウム論文集 (情報処理学会シンポジウムシリーズ, Vol.98, No.15), 情報処理学会, pp.25-32 (1998).
- 6) 高田広章, 田丸喜一郎: オープン化のためのリアルタイム技術と ITRON プロジェクトにおける取組み, システム/制御/情報, Vol.43, No.1, pp.34-41 (1999).
- 7) Sakamura, K. (Ed.): μ ITRON3.0: An Open and Portable Real-Time Operating System for Embedded Systems, IEEE CS Press (1997).
- 8) <http://www.itron.gr.jp/>

(平成 11 年 2 月 9 日受付)