

ユビキタス時代の ZIPC と NEC ソフト開発環境

日本電気株式会社

NEC エレクトロニクスデバイス マイクロコンピュータ開発事業部
システム部 プロジェクトマネージャー

橋本 一也

1. ユビキタス時代のシステム開発要件と開発ツール

近年、マイコン組み込み機器のソフトウェア規模や複雑さが増大傾向にあり、また商品化サイクルが年々早くなってきており、その対応が非常に難しくなっていると感じています。マイコン組み込みシステム開発を行っているお客様の一番の関心事は、「ユーザのニーズを先取りした商品をコスト安く開発し、いち早く製品化すること」であると言えるでしょう。

さらに最近では、セキュリティへの関心が非常に高くなっています。例えば、最近よく話題にあがる JR 東日本の SUICA カードや ETC などの交通課金システム、クレジットカードのように、金銭、またはそれに相当するものを取り扱うシステムはお金と同じ価値を有することから改ざんや偽造されると大変なことになってしまいます。また、住民基本台帳や携帯電話 SIM (Subscriber Identity Module) のように個人の大切な情報が漏れ、悪用されることは許されません。さらには、主にハードウェアやシステム全体に適

用されている PL 問題でもソフトウェアの観点での PL 対応への取り組みがあります(「ソフトウェア技術者のための PL 法ガイド」日科技連出版社を参照してください)。そして、ご存じのように、ユビキタス時代のシステムは、高度にネットワーク化された機器が通信を行います。この通信はセキュアでなければなりません。外部から盗聴されない、悪用されないシステムとならなければならず、安全性配慮が最も重要なシステム開発要件になることは間違いありません。

このような安全性に配慮したシステムを開発するためにはセキュリティ技術が特に重要になり、また、そのシステムを製品とするためには、ある認定機関による「認証」というお墨付きを取得することが必須条件となってくるものもあります。この認証を取得するためには、さまざまな要件に対応しなければなりません。如何に品質を保証したかを証明する品質記録(ドキュメント等)の整備も要件のひとつになってきます。

このようなソフトウェア規模や複雑さの増大、製品の短サイクル、安全性への配慮、といった課題に対応するためには、マイコン組み込みソフト開発環境（下流ツール）だけでは不十分です。中流や上流ツールが持つ高度な表現力やプロジェクト管理力の活用が不可欠です。しかしながらただ単独のツールとして使ってしまったら、「下位レベルの変更に従従できない」、「記述レベルが異なるのでデバッグが困難」といった不都合が出てきて、結局は中流・上流ツールはドキュメントツールになってしまい、もったいないことになってしまいます。やはり、上流から下流までの仕様/設計情報やそれを使ったツールの操作性などを一貫して管理/操作できるしくみが必要になります。つまり、上流や中流ツールと下流ツールを連動させ、製品開発の上流工程から下流工程までシームレスに作業でき、無駄な作業をできるだけ排除することにより製品性能品質向上/開発TAT 短縮/安全性品質向上できるような環境が必要になってくるわけです。

2. NEC ソフト開発環境

NEC では、V850 シリーズマイコンを中心に、78K シリーズマイコン、VR シリーズマイコン用のソフト開発環境をツールベンダー様の協力の基に整備を行っています。

NEC のソフト開発環境は「プラッ

トフォーム」「ベースツール」「先行ツール」の 3 つに大きく分類できます。これらのツール群は相互に連携し、各ツールが持つ機能を効率良く使うことができるように配慮しています。

NEC が提案するソフト開発環境は、

- 各ツールによりデバイスの性能を最大限に引き出します
- シームレスな環境により開発TAT が短縮できます
- 高品質部品の利用により品質を向上できます
- 拡張性があり、柔軟な対応ができます

といった特徴があり、お客様が安心して使える環境を目指しています。

プラットフォーム

「プラットフォーム」では、OS、ミドルウェア、デバイスドライバ、ボードを中心としたソリューションを提供します。お客様セットを素早く製造して評価/機能確認していただくためのものです。

OSはμITRON4.0やOSEKを中心に各 V850/78K/VR シリーズ用として品揃えを行っており、高い信頼性をもつNEC製OSを使うことにより、お客様セット全体の品質確保にも貢献できます。

ミドルウェアやデバイスドライバは、音声認識、TCP/IP、CAN/LINドライバなど標準的な機能を用意し、お客様

セットを短期間で開発できるように支援します。

SolutionGear は評価ボードであり、そのボード上にμITRON や各種デバイスドライバ、ミドルウェアを搭載し、V850/VR シリーズマイコン用の、すぐに使えるレファレンス環境として提供しております。

ベースツール

「ベースツール」は、マイコンの性能を最大限に引き出すコンパイラ（CA）や洗練された GUI を持つデバugg（ID）、また NEC 製マイコンの高性能化に対応したエミュレータ（IE）を中心としたツール群です。

さらに、周辺機能シミュレーションを充実したマイコンシミュレータ（SM）業界で最初に製品化したパフォーマンスアナライザ（AZ）、またそれを進化させ、最適なメモリ配置を自動的に決定する性能チューニングツール（TW）といった特徴あるツールも用意しています。

先行ツール

「先行ツール」は、現状のツール群にはない先端機能や、新しいコンセプトのツールです。その中のひとつですが、現在、検査の自動化を行なうことによりお客様の検査工数を大幅に短縮できる新ツールを具体化しています。このツールをコンセプトレベルで紹介

させていただいており、お客様にはご好評いただいています。開発完了しましたら、随時、製品化して行く予定ですので、ご期待ください。

3. 上流・中流ツールと連携した NEC ソフト開発環境

上述した NEC ソフト開発環境は、下流工程、つまり、製造や検査工程での使用を中心としたツールであります。これだけだと限度があり、先に述べたお客様のシステム開発要件に応えるには充分ではありません。

図1は、ソフトウェア開発におけるエラー原因をあらわしています。エラー原因の半分以上は仕様と設計のミスであることがわかります。つまり、システム全体の品質を確保するためには、中流や上流ツールを活用し、もっと前段階での品質確保が必要であることを示しています。

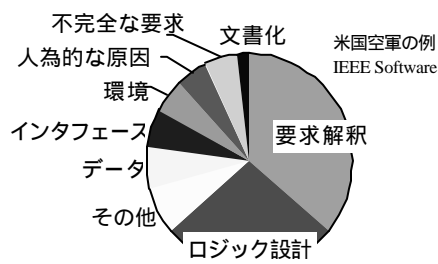


図1 エラー原因

NEC では、お客様のソフト開発作業がスムーズに流れるよう、この中流・上流ツールと下流ツールである NEC ソフト開発環境をシームレスに連動させるように整備しています。

図 2 は、NEC ソフト開発環境と中流・上流ツールを連動させたソフト開発環境全体の構成図です。各ツール群は ToolLink というインターフェースで接続しています。ToolLink はマイコンソフト開発ツール間連携インターフェースであり、NEC など半導体メーカや

キャッツ社をはじめとした各ツールベンダー様が標準化を推進しており、業界標準を目指しています。詳細は「ZIPC WATCHERS Vol.5」で紹介されていますのでご参照ください。

(ToolLink ホームページ :
<http://www.semicon.panasonic.co.jp/micom/debugfactory/toollink/index.html>,
 メールリングリスト :
tisc-all@mctcs5.ucom.lsi.nec.co.jp)

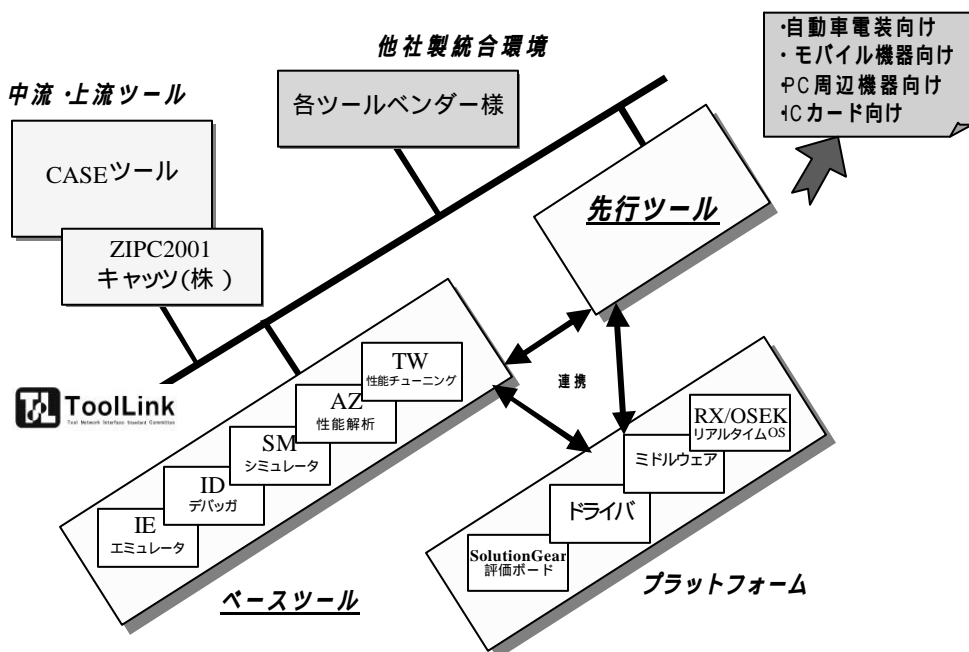
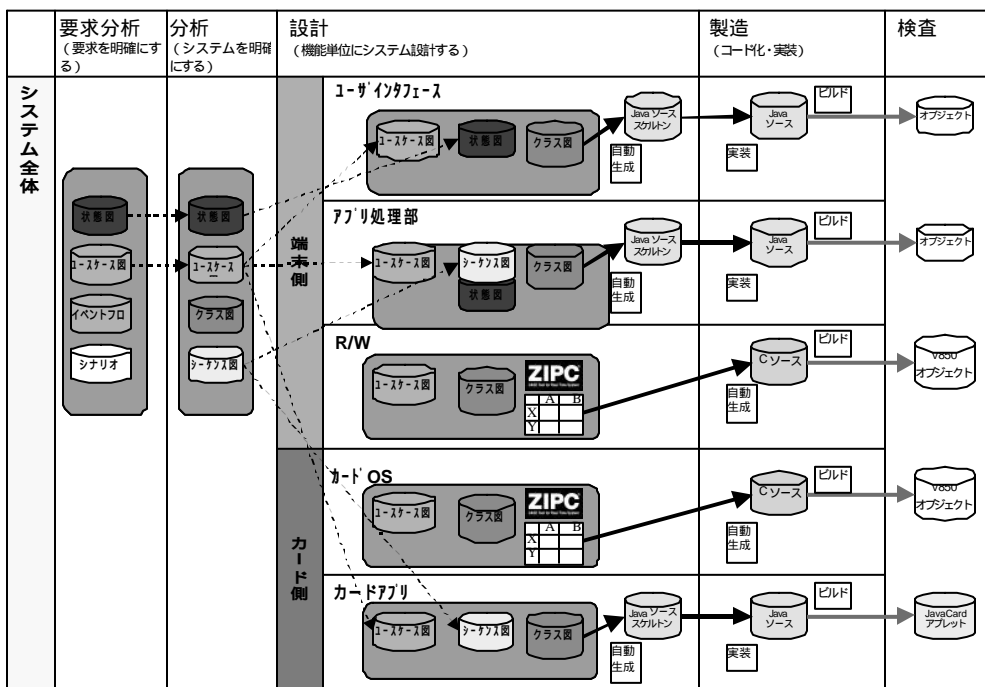


図 2 NEC ソフト開発環境 (全体構成)

ツール間の連動が可能になると、例えば、ZIPC 状態遷移表からブレークポイントを設定したり、セル単位の実行など実行制御ができ、つまり、中流や上流ツール上で実機デバッグが可能となります。そして、NEC では、これら上流から下流までのツールを、ツールチェーンとして各マイコン応用分野別に紹介しています。ここでは、ICカード向けツールチェーンと、自動車電装向けツールチェーンを例にして説明します。

マイコン応用分野別に紹介している

理由は、自動車電装系では MATLAB 系が、携帯電話系は SDL が比較的良好に使われている、と言ったように各分野において、ソフト開発プロセス、設計スタイル等が異なって、使うツールも異なるからです。また、システムの中でも各機能により、使うツールや図面（シーケンス図やクラス図など）が異なってきます。図3は、ICカードシステム全体の開発プロセスの一例です。設計段階になると各層に分けて作業を行います。



Empower your creativity
V850
Emulation Controller

図3 Java カードシステム開発プロセスの例

IC カードシステム開発環境

表1は、伝送プロトコルを ZIPC で記述したものです。IC カードと端末の間の通信を ZIPC を使って実装してみました。IC カードでは、通信規格は ISO で規格化されており、その仕様は、状態遷移表で公開されていますので、この時点で ZIPC とは相性が良かったわけです。

ZIPC を使って実装した結果、NEC 製 IC カードマイコンを搭載した Java カードでは問題なく動作しました。ZIPC と NEC ソフト開発環境の連携で、状態遷移表レベルで実行カバレツ

ジを取り、仕様通りの動作保証もできます。

この例は、ZIPC を使うことにより仕様通りにシステムの動作保証ができるということを示しています。

今後 IC カードシステムの市場規模は大きくなると予想されておりますが、安全性が一番要求される分野であり、そのシステム開発には安全性への配慮が必要になってきます。NEC ではその安全性に配慮した IC カード用マイコンとそのソフト開発環境を提供していきます。

表1 ZIPC で記述したISO7816 伝送マトリックス(実例)

ICカード状態	初期状態	受信待ち状態	リトライオーバー	処理中	コマンド応答待ち状態
I正常	受信シーケンス番号更新; コマンド解釈開始; リトライカウンタ初期化	受信シーケンス番号更新; コマンド解釈開始; リトライカウンタ初期化	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
	リトライ(ERR); sleep	リトライ(ERR); sleep	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
II正常	リトライカウンタ更新; HOST送信(ERR2);	リトライカウンタ更新; HOST送信(ERR2);	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
WTX_RES	リトライ(ERR); sleep	リトライ(ERR); sleep	受信待ち処理;	コマンド解釈開始; リトライカウンタ初期化	
受信エラー	受信シーケンス番号初期化; 送信シーケンス番号初期化; リトライカウンタ初期化; HOST送信(再送信);	受信シーケンス番号初期化; 送信シーケンス番号初期化; リトライカウンタ初期化; HOST送信(再送信);	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
	リトライ(ERR); sleep	リトライ(ERR); sleep	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
EDCエラー	リトライカウンタ更新; HOST送信(ERR1);	リトライカウンタ更新; HOST送信(ERR1);	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
デレスエラー	受信待ち処理;	受信待ち処理;	受信待ち処理;	受信待ち処理;	
異常コマンド	リトライ(ERR); sleep	リトライ(ERR); sleep	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
	リトライカウンタ更新; HOST送信(ERR);	リトライカウンタ更新; HOST送信(ERR);	受信待ち処理;	リトライ(ERR); sleep	リトライカウンタ更新; HOST送信(WTX);
伝送エラー	Iブロック				送信ブロック回収; 送信シーケンス番号更新; 送信データ保存; HOST送信(I);
	WTX_RES				HOST送信(WTX);

車輻内ネットワークシステム開発環境
 車輻内で使われる ECU (電子制御装置) 中の OS として OSEK が注目されています。OSEK は欧州で標準化された OS です。また、車輻内で使われる ECU (電子制御装置) 間の通信には CAN や LIN 等の規格が採用されています。

図 4 は、車輻内ネットワークシステムの開発環境の例です。

NEC では CAN ネットワークのためのツールとして、OSEK カーネルや CAN バスアナライザ (SE-70000) を製品化し、またさらに、いち早く ZIPC の OSEK/CAN 対応を実現しました。具体的には、ZIPC の中で OSEK カー

ネルや OSEK/COM の API 関数 (システムコール) を発行できるようにしていただき、また、ZIPC 単体で CAN を介して通信できるように機能アップしていただきました。これにより、設計時のモレ・ヌケのチェック、エミュレータなしでの動作確認、また ZIPC で記述した簡易的な ECU 仮想ターゲットとして利用できるようになります。

このように、NEC では各分野毎に有効となるようなツールチェーンの整備を行ない、効果を実証した上でお客様にソリューションとして提案しております。

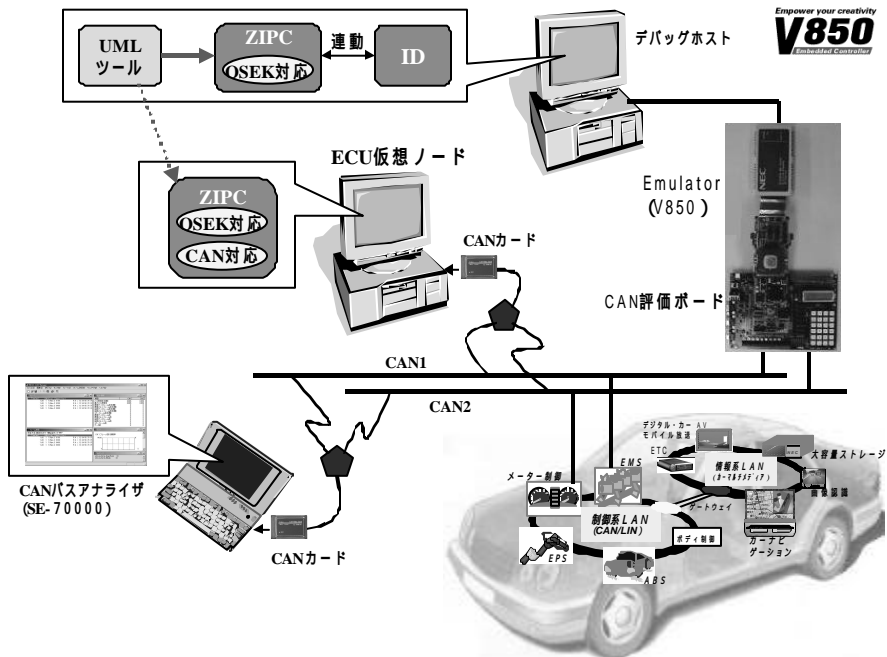


図 4 車輻内ネットワークシステム開発環境

4. これからの開発環境

先にも述べたように、ユビキタスの時代になると、各機器に通信機能が装備されネットワークを形成することになってきます。上述の IC カードシステムや車輦内ネットワークシステムはユビキタスコンピューティングを具現化したもののひとつと言えるでしょう。

この基本機能の通信と言えば、“状態遷移”が活躍できる場面です。先に挙げた IC カード伝送プロトコルの ISO や ITU などで規格の表現として状態遷移表が使われていますので、この場面では ZIPC が一番適していると言えるでしょう。

今まで Excel などの表で状態遷移を表し、仕様書として使うことが多かったと思いますが、ZIPC や Konesa-RealTime の状態遷移表を使えば、状態遷移表レベルでシミュレーションができたり、容易にデバッグをすることができます。しかも、ZIPC 単体で表レベルのシミュレーションができ、上位のレベルでバグを早い段階で除くことにより、検査工程で細かいデバッグ作業をすることなく、システム全体の品質を高めることができます。

ZIPC や Konesa-RealTime は、その状態遷移表が持つモレ・ヌケのチェック機能を十分に活かすことにより、ユビキタスの時代でも重要なツールとして位置付けられると思います。

NEC では、キャッツ社をはじめ強

力なツールを持つツールベンダー様と連携し、ユビキタス時代の新たなお客様の価値創造に貢献できるようなソリューションを提供していきたいと考えています。

- Suica は JR 東日本の登録商標です。
- MATLAB は The MathWorks 社の製品です。
- Rose はラショナルソフトウェア社の製品です。
- ZIPC および Konesa-RealTime は キャッツ社の製品です。
- Java およびすべての Java 関連の商標およびロゴは、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。