～ＺＩＰＣは小規模ハードリアルタイム制御に使えるか？～

松下電器産業株式会社　ソフトウェア開発本部
ソフトウェアエンジニアリンググループ　技師
古山　寿樹

1．はじめに

弊社で製造する家電製品における組込み制御ソフト（以下、組込みソフト）の役割は、近年急激に増大する一方、品質の確保や生産性向上への要求は日に日に厳しさを増している。そこで、

●状態遷移表設計による設計完成度の向上
●コード自動生成による開発効率化
●VIP機能によるハードに依存しないソフト機能シミュレーションの実現

等の効果を期待して、実AV商品（CPU：８ビット、ROM容量：４８K）をモデルにZIPC Ver.5.0の導入検討（評価）を行ったのでその結果について報告する。

2．組込みソフトの特徴

家電製品の組込みソフトには以下の特徴がある。

（1）自律的な連続動作

外部イベントであるユーザからの「START」キーの入力で動作を開始し「STOP」キーが押されるまで、メカやハードの状態に応じて自律的に動き続けるものが多い（商品によっては電源投入により動き始め、電源が供給される間、動き続ける必要がある）。

（2）リアルタイム性の確保

商品そのものの高機能化に対応した制御及びデータ処理の高速化に加え日常的に使う家電製品については、安全性や快適性を確保するためのシビアなリアルタイム処理が要求される。

（3）コストダウンのための厳しい制約

市場価格の低下に伴いコストダウンの要請は厳しく、安いリソース（遅いCPUと少ないメモリ）で高機能の実現が求められる。

3．評価項目

状態遷移表の考え方そのものは、システム内の主要（複雑）な状態遷移を正確に表現する手段の一つとして従来から活用されてきており、その有効性は認識されている。今回は特に現状業務に対して開発ツールがどの程度適用可能であるかを中心に以下の項目について評価を行った。

●状態遷移表設計の適用範囲は？（有効な使い方は？）
●自動生成コードの使用可能性は？（速度、サイズは？）
●シミュレーション（VIP）機能によりハ

ードレス開発は可能か？

4．状態遷移表設計の適用範囲

(1) 状態遷移表を用いた要求仕様分析

本来ZIPCは、設計のためのツールであるが、既存システムの分析には非常に有効なツールである。既存のソースを読み状態遷移表形式で記述する事によって、システム内の全ての動作を明らかにすることができる。特にZIPCの適用においては日本語による状態遷移表の表記が可能なため開発者以外の技術者による仕様確認やレビューも可能となる。

ただし、システム全体の動作（状態遷移）の全てを状態遷移表で記述しようとすると、表の規模が大きくなるため、作成に時間がかかり視認性も落ちる。適用の際には要求仕様分析に入る前に実現する機能範囲を十分に絞り込む必要がある。

(2) 状態遷移表による設計

状態遷移表の考え方の基本は「イベントドリブン」にしろ「状態ドリブン」にしろ、その遷移には必ずイベントが必要である。シーケンス処理の多い制御への適用には状態の抽出や処理の表現に工夫が必要となる。

例えば、研修テキストなどのオーディオ機器を例にした状態遷移表では、必ず「再生中」や「停止中」と言った状態が登場する。しかし、実際の制御においては、多くのシーケンス（自動遷移）処理を含む多くの複雑な状態が存在し、「再生中」と言った抽象的な状態は存在しない。又、「停止中」状態についても停止に至る過程や停止後の動作に依存した複数の状態が存在するのが普通であり、何を状態にするのか、イベントの無い自己遷移をどう表現するか等の設計方針と表記上の工夫が必要となる。

5．ZIPCによる実装（自動コード生成）

ZIPCによる状態遷移表設計のメリットは自動コード生成による開発の効率化である。しかし、実開発においては制約条件（サイズや速度）とのトレードオフが発生するため、ツールの使用ノウハウやチューニングが必要となる。

(1) サイズの制約

状態遷移表設計により、制御処理がセル単位で整理され、冗長なデータの排除、判断処理の一元化、共通関数化等による共通処理部分の削減によるコード削減が期待できる。しかし、ZIPCによる自動生成では状態遷移表の実行処理部分がサイズオーバーヘッドとなりコードサイズが増加する。特にZIPCでは状態遷移表をテーブルとして実装するため、コードサイズは設計する状態遷移表に比例して大きくなる。従ってコードサイズを意識すると状態数は増やせない。（実用サイズは１状態遷移表当り最大１６×１６）

又、商品の特性を考えると「S（状態駆動）型」を選択すべきだが、コードサイズを考慮すると「E（イベント駆動）型」しか利用することができない。

≪生成オプションによるオーバーヘッドサイズの比較≫

STM 種別	S 型通常			E 型通常		
ｽｹｼﾞｭｰﾙ型	S 型			E 型		
ｲﾍﾞﾝﾄ解析	状態毎解析	状態毎解析		ﾙｰﾄ解析実行		
	STM 単位	処理単位：処理のみ				
ﾏﾄﾘｯｸｽﾃｰﾌﾞﾙ	標準型			処理抜粋型	ｵﾌｾｯﾄ型（char）	ｵﾌｾｯﾄ型（short）
同一処理共通化	処理／遷移					
ｵｰﾊﾞｰﾍｯﾄﾞ率	0.62	0.56	0.56	0.30	0.39	0.53

オーバーヘッド率 ＝ オーバーヘッドサイズ／全生コードサイズオーバーヘッドサイズ
＝ ﾒｲﾝ関数＋ｲﾍﾞﾝﾄ解析関数＋ｲﾍﾞﾝﾄｱｸﾃｨﾋﾞﾃｨ関数＋遷移関数＋遷移関数
＋ZIPC RAM 変数＋STMﾃｰﾌﾞﾙ情報
＝ 全生成コードサイズ－実行関数サイズ

（2）速度の制約

以下の理由により遷移表実行時の１サイクルの遅れが致命的な動作遅れになるような速度要求の厳しいシステムでは自動生成コードはそのままでは利用できない。これを回避するためには自動生成されたコードをチューニングする必要がある。

① 最もサイズオーバヘッドの少ない処理抜粋型ではセル番号の検索処理のため処理時間が１サイクル時間に大きく影響してしまう。

② 現状のイベント解析処理では複数のイベントが同時に発生した場合、非実行セル（不可、無視）も、イベントが発生していればイベントチェックが行われる。更に処理のないセルが優先順位が高い場合、それ以降の判断は行われない。

(3)コード品質（ＩＦ文は減るか？）

状態遷移表による設計では IF 文を減らす効果があり、この事によりソフト品質の向上とサイズの圧縮が見込まれる。

しかし、実際の商品仕様ではイベントのみで動作が確定する事は少なく、イベント発生時の状態（通常複数の状態の組合わせ）を判断する必要がある。

イベントを増やさずに状態を増やす方法としては状態変数を用いることになり、アクションセル内には IF 文が多く表われてしまうことになる。IF 文の数そのものは減らせないが、条件判断のためのセルと動作のセルを分離し条件判断処理を集中させることにより IF 文の分散を押さえることはできる。

（4）デバッグ容易性

本来、ソフトウェアの自動生成においては、実装コードを意識しない開発が理想的であるが、組込みソフトの開発においては､単純なソフトの問題だけではなく、ハードや外因の影響による動作不良を細かく検証する場合もあり、デバッグ工程は必要不可欠なものである。

効率的なデバッグのためには開発環境そのものの機能に加えてコードの読みやすさも必要である。現状の自動生成コードは、コメントもほとんど無いため、ZIPC独自の実装アーキテクチャを理解する必要がある。

6．ZIPCによるシミュレーション

今回の評価ではVIPを用いたソフト上での動作シミュレーションの検討も行った。今回の開発システムでは、近年のシステム開発ではLSIとの並行開発が多いため、開発初期からシミュレーションが可能になれば効率的なハードレス開発が実現できる。ユーザ入力そのものの数はそれほど多くないため簡単にシミュレーションを行うことができる。しかし、ハードの無い状態でソフトウェアの動作確認を行うためにはLSIの動作そのものもシミュレートする必要がある。特にプロトコルを持つシリアル通信などのLSIとの通信処理ではタイムアウトも大きなエラー要因になるため、人手による数値設定やボタン操作を用いた動作実行には限界がある。実適用のためには、容易にＬＳＩの動作そのものシミュレーションができるような仕組みが必要となる。

7．評価結果のまとめ

今回の導入目標であった３つの項目について以下にまとめる。

（1）状態遷移表設計による設計完成度の向上

状態遷移表設計の導入により、ソフトウェア動作モデルの可視化が可能となり、モレ・抜けは大幅に削減されると考えられる。又、日本語による表現が可能なため、設計当初のシステム分析やレビューには非常に有効である。しかし、設計への適用においてはシーケンス処理に対する表現上の工夫が必要である。

（2）コード自動生成による開発効率化

サイズや速度の制約が厳しいシステムへの適用には生成後のコードのチューニングが必要であるため実用的ではない。ある程度の規模以上のROM容量に余裕のある大きなシステムへの適用には効果があると考えられる。

（3）ハード依存性の低いソフトシミュレーション

人間が行う機器操作のシミュレーションは簡単に実現することができ非常に有効

であるが、LSI のような速度が早く信号の多いハードウェアをソフト環境のみでシミュレーションするためには、対象商品専用のシミュレーション機能を開発する必要がある。

　又、今回はオブジェクト指向分析との連携についても評価したが、この内容については紙面の都合により別の機会に紹介したい。

ZIPC Watchers Back Number
のお知らせ

http://www.zipc.com/

本誌ZIPC Watchersのバックナンバーダウンロードサービスを行っております。

創刊号からVol.4までの全内容をPDF化。弊社Webサイトよりダウンロードできます。

創刊号から通して読むことでZIPCの歴史とともに、業界の動きが手にとるようにわかります。

★ ***ZIPC Watchers Vol.1*** (1997年　3月25日初版発行)　B5版／34頁
★ ***ZIPC Watchers Vol.2*** (1998年　6月19日初版発行)　B5版／53頁
★ ***ZIPC Watchers Vol.3*** (1997年　5月28日初版発行)　B5版／77頁
★ ***ZIPC Watchers Vol.4*** (2000年　6月16日初版発行)　B5版／80頁

※　記載されている、寄稿者様のご所属部署は当時のものです