モーター制御プログラム開発へのZIPC適用
－ZIPC導入に際しての評価結果を踏まえて－

中央電子株式会社
制御システム事業部
船田　晋

以下にCASEツール「ZIPC」の評価を報告する。今回は、10軸モータ制御プログラムを参考に、機能ごとに定性的な評価を行った。

１．評価対象

「キャッツ株式会社」製　ZIPC 2001 Version 8.0 Professional Model

・エディタ機能、チェッカー機能、シミュレーション機能、ジェネレータ機能、VIP機能、ATV機能、リバース機能、その他

今回は、主にエディタ、チェッカー、シミュレーション機能についての評価を行った。

【参考文献】渡辺政彦著、拡張階層化状態遷移表設計手法Ver.2.0

２．参考プログラム

以前に製作したプログラムを参考にする。これは、RTOSを使用しない割込み処理主体のモータ制御プログラムであり、10軸のモータを2ch、５軸ずつ制御し、位置合わせを行うプログラムである。

No	項目	実際の処理内容	今回の参考内容
1	ソレノイド制御	上位からの指令により、ソレノイドの制御を行う。	処理内容は省略し、遷移の流れだけを参考にする。
2	モータ回転シーケンス	上位からの指令により、回転パルス数を計算し、モータ停止→回転→停止、また緊急停止のシーケンス処理を行う。	処理内容は省略し、遷移の流れだけを参考にする。
3	125us周期の割込み	ＤＩの読込み処理とモータ駆動処理を行う。	周期ハンドラで DI を監視し、イベントを起こす。

３．評価結果

3.1．所要日数

上記のプログラムを参考にして、状態遷移表（以下、STM）を作成したところ、下表の日数を必要とした。今回、ツールの利用に不慣れなところもあるので、今後の活用時においてはさらなる短縮が可能である。

表1：作成できる設計書

No	ドキュメント種類	作業内容／用途	所要日数
1	状態遷移表（STM）	イベント／状態抽出	２日
		表作成	７日
2	状態遷移図	状態遷移表からコンバートする	
3	シーケンス図	STM 作成時に必要(STM とは独立している)	
4	タイミング図	STM 作成時に必要(STM とは独立している)	

その他のドキュメントとして、関数定義書、変数設計書、定数設計書があるが、これらは今までヘッダーファイルにコーディングしていたものを個別に記述したものと考えてよい。

3.2．エディタ機能

別添資料１は、プログラムのメインフロー～ソレノイド制御～モータ回転シーケンス部分のフローチャートを示している。これをSTMで表すと、別添資料２のようになる。状態を複数のフラグで分けていたものが、遷移表にしたことで、実際の処理内容は変わらないにも関わらず、視覚化、単純化することができる。

実装を意識せずに設計ドキュメント（状態遷移表や状態遷移図など）としてのみ扱うのであれば、ソレノイド制御、割り込みなどを記述するのは容易である。

実際にSTMを作成したところ、エディタ機能の具体的な特徴として以下の点が挙げられる。

（１）STMの表記法、あるいは操作の習得は、ZIPC初心者でも容易である。ただし、状態とイベントの抽出には、要求仕様と十分に照らし合わせて考え、また処理がイベント駆動型であるか状態駆動型のどちらが好ましいかをよく検討することが必要である。

（２）イベント駆動型のSTMの場合、イベントが起こらないと処理に進まないので、シーケンス（自動遷移）処理をする場合の記述には、処理内にイベントをコールする関数を追加するなど、少々の工夫が必要となる。

（３）セル内に処理を記述して、プログラミングすることも可能であるが、それでは実装段階に踏み入ることになるので、なるべく設計をしているという意識で記述するのがよい。（セル内のＩＦ文を減らして、「状態」側を条件分岐するなど）

（４）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有効である。また、ハードウェア技術者が容易に理解しやすくなるという利点も考えられる。

（５）イベントや状態の開始時、終了時の処理を、数種類のマークを用いて表しているので、視覚的に遷移の流れを追うことができてよい。

（６）モータの５軸制御の場合は、５軸分のSTMで実現するのが一般的である。また全てのSTMにおいて、一旦イベントを受け付けた後、状態遷移表処理が終了するまでの間、次のイベントを受け付けることはできないようである。数軸制御のサンプルがほしい。

（７）周期ハンドラについては、特別な作りこみが必要となる。割り込みハンドラを仲介してイベントを受け付けるのが一般的である。ハンドラでタイマー割り込みを受け付け、ハンドラからSTMにイベント（メッセージやフラグなどの方法にて）を送る。この辺りの設計のヘルプ機能をもう少し充実させてほしい。

（８）STMからコンバートした状態遷移図は、線が絡み合い見にくく、設計書に添付するドキュメントとして使用するのは難しい。

（９）印刷機能は、STMがすべて印刷範囲に入るようにすると、テキストが重なり、非常に見づらくなる。長い処理文を挿入する際には省略するなどするしかない。（８）と合わせて、何らかの添付ドキュメント用の設定が必要である。

3.3．チェッカー・シミュレーション機能

（１）ZIPC最大の特徴であるモレ・ヌケのチェックであるが、処理がセル単位になっており、すべてのセルをチェックすることで、非常に発見しやすくなっている点はよい。

（２）イベント発行によるシミュレーションの実行が可能である。各イベント、状態、セル内にBREAKを置いて、遷移の流れを追う操作は問題なくできる点は素晴らしい。

（３）子STMを複数作成した場合には、シミュレーション時にアクティブになっているセルも複数あり、把握しづらい面もある。特に、割込み処理を挿入した場合、アクティブ画面が頻繁に移動し、画面が「チカチカ」するので、モード選択等をして改良してほしい。

3.4．ジェネレーション機能

（1）自動生成コードは、状態遷移表の１セルを単位として生成するので、あまり「状態」の数が多いと、生成コードのサイズも大きくなるようである。上限のサイズを選択してSTMを最適化するような処理がほしい。

（2）リバースエンジニアリング機能はあるが、初めにZIPCで生成したコードを編集したものをリバースして状態遷移表やその他のドキュメントに反映するものである。ZIPCによらない既存のコードから状態遷移表を生成するリバース機能があるといい。

3.5．ＶＩＰ機能

（1）VCまたはVBで作成した擬似ターゲットとの関連付けには、手間のかかる設定が必要であるが単純な作業ででき、シミュレーションとの連動を行える点は素晴らしく、また設計・製作を楽しくするという面でも画期的であると感じた。

（2）開発の初期段階からシミュレーションが行えるという利点がある。ハードウェア担当者と共に仕様を確認しあうことが一番のメリットである。

（3）「VIP環境設定」の「名称イベント設定」タブの削除ボタンが有効にならないことがあった。サンプルではなく、詳細な説明書がほしいところである。

3.6．その他の機能

今回は評価の対象外であったが、その他の機能を簡単に紹介する。

（1）ATV機能・・・作成したSTMを基に、すべてのイベント、状態の組み合わせについてパターンを作成して、自動試験、検証することができる。

（2）ターゲット・デバッグ機能・・・HEW等のツールと連携し、ターゲットを接続してデバッグをする。

4．まとめ

今回は、ZIPCの機能をSTM作成～シミュレーション実行を中心に評価した。仕様の全体を再現するにはさらなる詳細な設計が必要であるが、様々なシステムに対応する詳細設定が可能となっており、さらに大きなシステムを設計する場合に必要な機能は、十分に揃っている。また、ハードウェアとのインターフェイス部分をどの程度まで表記できるかは、未知である点が多く、今後の試用に任せるが、他社の運用例からしても十分な対応ができると考えられる。

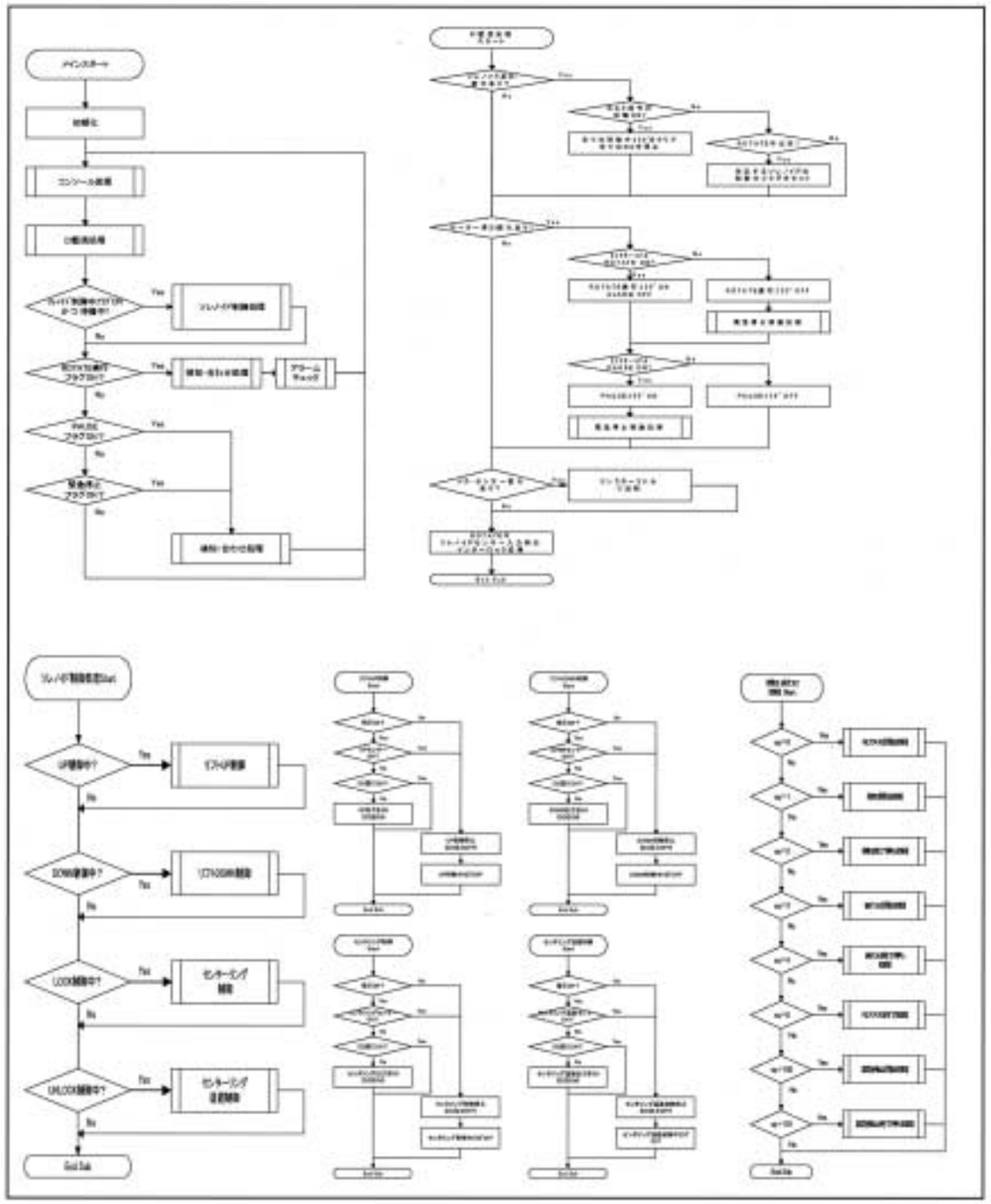
キャッツ（株）のサポート体制はよく整っており、また、ユーザーからの声を取り入れて改良している点も評価でき、実際の運用にはサポートをいただけるものと考える。

部内業務については、一つモデルが完成すれば、若干の変更をすることで、他システムに流用しやすいという点もある。

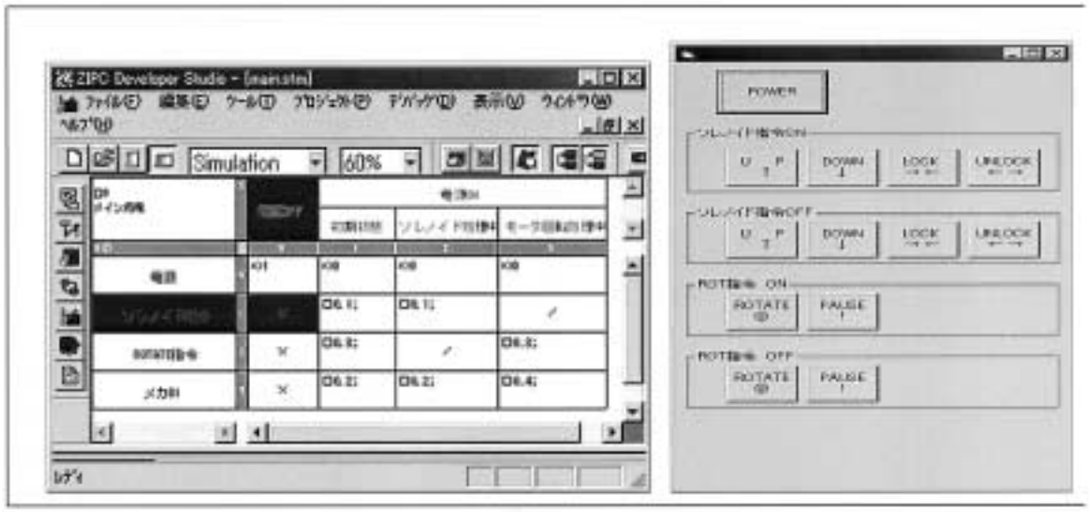
以上の評価結果を考慮すれば、組み込みシステムへのＺＩＰＣの使用は有効である。今後の組込みシステムの複雑化を考えれば、制御システム事業部の業務において、品質向上、開発期間削減のためには、ＺＩＰＣは必要なツールであると考える。

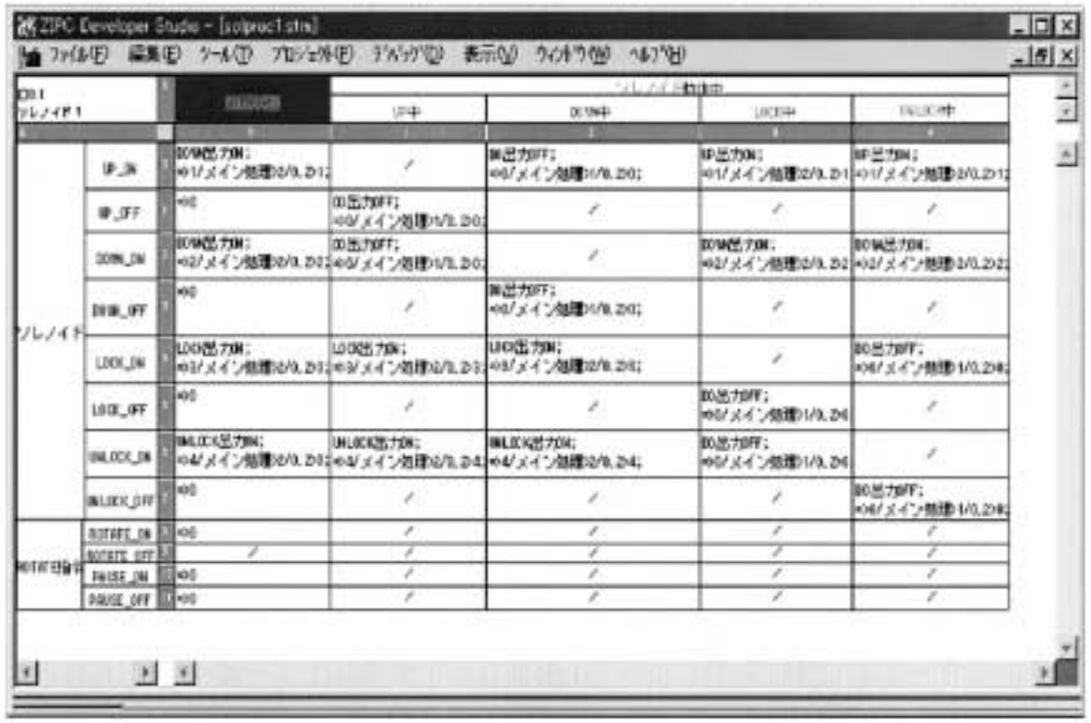
以上

資料１　メインフロー～ソレノイド制御～モータ回転シーケンスのフローチャート



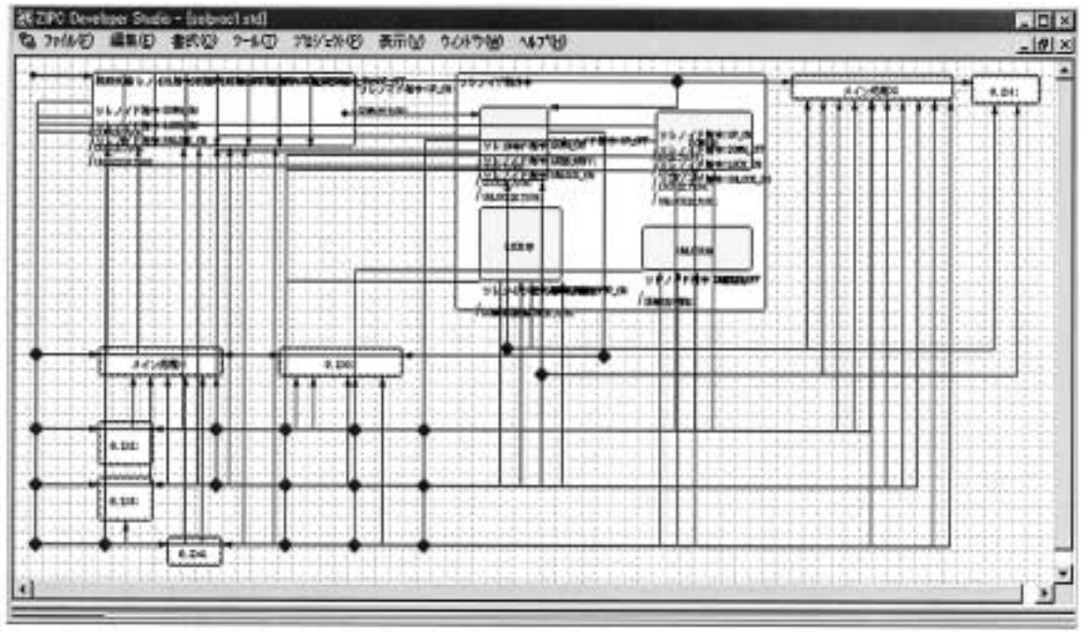
資料２　状態遷移表による設計





ZIPC WATCHERS Vol.8

資料３　STMからコンバートした状態遷移図の例（ソレノイド制御部分）



※状態遷移図のエディタ機能は充実している。ただし、イベントや状態の数が多いと、文字が重なり
　非常に見にくくなるので注意が必要。