

車載ソフトウェアのテスト自動化支援ツール

片岡 智美*・坂 育子・古戸 健
松本 達治

Test Automation Support Tool for Automotive Software — by Tomomi Kataoka, Ikuko Saka, Ken Furuto and Tatsuji Matsumoto — In recent years, automotive components have become more sophisticated and the electronic control unit (ECU) has employed more complex large-scale software. As the product scale becomes larger, an increasing number of tests are required to assure product quality. Even in the case that auto-testing tools are used, test patterns need to be input manually. This process requires significant amounts of man-hours particularly when the product types vary and the test patterns need to be modified for input signal changes. In order to improve the efficiency of this product development process, we have developed a tool that converts the simulation patterns used in model-based design into those for product tests. This tool automatically adjusts the input signals, and thus, successfully reduces the man-hours by 50% and improves the test quality with common test patterns.

Keywords: software, test method, model based development

1. 緒言

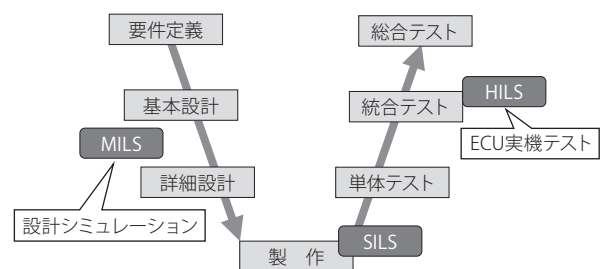
当社では、自動車に搭載される電子制御ユニット（以下、ECU*1）を開発している。ECUのソフトウェア（以下、車載ソフトウェア）は、全く新規に設計されることは少なく、多くの場合、旧モデルの設計を流用して開発される。設計の流用時には、車載ソフトウェアの品質確保のためのテストパターンも基本的に流用する。ところが、車両全体設計が見直され複数の車載ECUへの機能分割、接続インタフェースの変更などが行われることがある。この場合、自動車として提供される機能に変更がなくても個々のECUには信号入力・出力インタフェースの設計変更が生ずるため、ECUごとにテストパターンの再設計が必要という課題があった。

一方、自動車の高機能化により車載ソフトウェアは大規模化・複雑化の傾向にある⁽¹⁾。テスト項目数の増大、検証モレのリスク増大などの問題解決のため、モデルベース開発手法を自動車業界では導入推進している⁽²⁾。モデルベース開発では、設計図（モデル）をシミュレーションすることで、設計を検証する（MILS*2）（図1）。この手法により、設計品質が向上し、テスト工程でのバグ発見と手戻りの減少という効果がある⁽³⁾。さらに、設計シミュレーションで用いるテストパターンをテスト工程で用いるテストパターンと共通化すれば、開発全体が効率化できる。しかし、モデルとテスト工程（HILS*3）での検証対象であるECUとでは信号入力・出力インタフェースが異なるため、信号入力から出力に至るタイミングも異なり、単純に共通化できないという課題があった。

本稿では、まず1つめの課題である車載ソフトウェアの

流用開発時の品質確保の効率化について述べる。そこでは、ECUの入出力インタフェースの設計変更前と設計変更後でテストパターンを共通化する技術に取り組んだ。

次に、2つめの課題であるモデルベース開発の効率化では、前記のテストパターン共通化の技術を応用して、設計シミュレーション用のテストパターンとECU実機テスト用のテストパターンとの共通化に取り組んだ。



	MILS	SILS	HILS
ECU部品	モデル	モデル	実機
制御部	モデル	プログラム	プログラム
車両	モデル	モデル	モデル

図1 モデルベース開発プロセス

2. ECUの流用開発時の品質確保の効率化

2-1 テストパターンの再設計に伴う課題 車載ソフトウェアの流用開発時に、旧モデルからの機能的な変更がなくてもテストパターンの再設計が必要なケースについて説明する。

図2に、元は1つのECUで提供していた機能を複数のECUへ分割配置した例を示す。(a)の構成では、スイッチ“SW_X”の入力を受け、負荷“RLY_Y”を制御するまでの一連の制御を1つのECUで実現している。一方、(b)の構成では、ECU-1で“SW_X”の入力を受け、ECU-2で“RLY_Y”の出力を制御するように役割を分ける構成としている。このように、複数のECU間を通信で多重化することにより車両全体の電線長を短くしてコストを削減できる。

ECU構成が変更された場合でもテスト設計を効率的に行うため、同一機能のテストは、構成1のECU用のテストパターンを、構成2のECU-1およびECU-2に流用したい。このとき、図2の(c)(d)に示すように、個々の入出力信号の種別(ジカ線^{※4}、あるいは、通信)や、信号の有無など、インタフェース部に違いがあるため、テストパターンの多くの関連箇所を変更しなければならない。

また、図2の(e)(f)で、構成1のECUと構成2のECU-2のテストパターンの例に着目してスイッチ入力のタイミングを比較する。これは、入力インタフェースがジカ線入力からCAN^{※5}入力に変更されたケースである。

一般的に、ジカ線からスイッチ入力の情報を取得する場合、ソフトウェアでフィルタ処理が行われる。フィルタ処理とは、信号の状態を定期的に観測し、あらかじめ定義した時間を超えて同じ状態が継続した場合にソフトウェア内部の入力情報として確定する方式で、スイッチ接点のチャ

タリングを除去する目的で車載ソフトウェアに組み込まれている。そのため、構成1の場合は、入力操作と同時のタイミングではなく、入力操作から前述のフィルタ処理時間(図ではTa)経過した後に、ユーザのスイッチ操作の状態はソフトウェアの内部情報として確定する。

一方、構成2のECU-2のように、通信でスイッチの入力情報を受け取る場合は、ECU-1側で既にフィルタ処理を実施済みのため、ECU-2側ではフィルタ処理を行わない。そのため、信号入力を受けてから制御信号を出力するまでの時間が異なる。この理由により、機能に変更がなくても、構成1のECUのテストパターンから、構成2のECU-2のテストパターンへは、先に述べた入出力信号の種別や、信号の有無などの入出力インタフェース部の変更に加えて、入力操作から制御出力までの経過時間も調整する必要があった。

これらECUへの入出力インタフェースや入出力タイミングの違いに対応するためには、個別に手作業でテストパターンを作り直す必要がある。テスト実施の効率化を図るため、市販のテスト自動化装置を導入した場合でも、テストパターンの作り直しは必要である。特に、当社が開発対象としているボディ制御^{※6}用のECUソフトウェアには、多くの入出力信号があることから、入出力インタフェース部の変更に伴う影響範囲は大きくなる傾向があり、テストパターン修正の作業量が課題となっていた。また、人手による作業が多く発生することから、テストパターン作成時に人為的なミスが入り込む可能性があり、テスト品質の面でも懸念があった。

2-2 入出力信号割付変換ツールの作成 そこで、2-1で示した課題に対応するために、元となるテストパターンに対して、信号名や入力種別、入力操作タイミングなど、入出力信号の割付に伴う変更点を自動的に変換する入出力信号割付変換ツール(以下、割付変換ツール)を開発した。

具体的には、図3(a)に示すような、入力情報と、その入力に対してソフトウェア内部で入力情報が確定するタイミングを同じタイミングとした場合の「基準テストパターン」を事前に作成しておく。この「基準テストパターン」を、前述の割付変換ツールで、入力種別に応じて、例えばジカ線入力タイプの場合は、図3(b)に示すように、入力フィルタ処理によりスイッチ操作時点からソフトウェア内部で確定するまでの時間Taを遡って、「基準テストパターン」より早いタイミングで変化するテストパターンを自動的に生成する。

また、この割付変換ツールには、幾つかの変換用のパラメータを持たせて、テストパターンに含まれる信号毎に個別に設定できるようにした(表1)。パラメータには、満たすべき機能が同じ場合でもテストパターンの差異として表れる、信号名の変更、通信とジカ線の種別の変更、変化タイミングの変更、および信号の削除等を行うための項目を

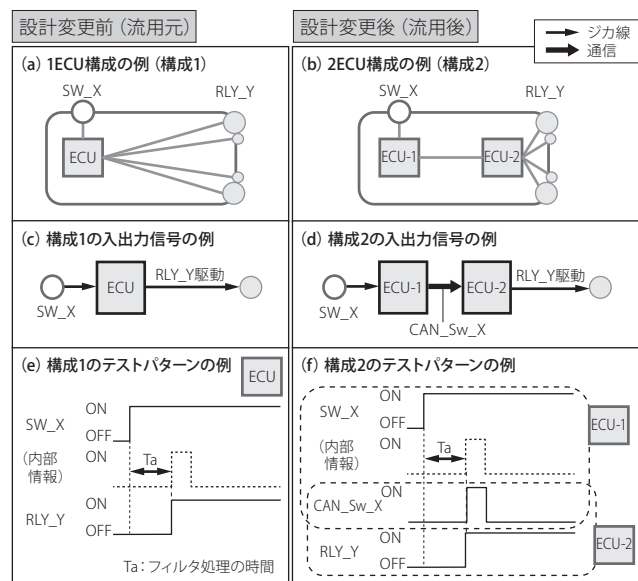


図2 ECU構成とテストパターンの例

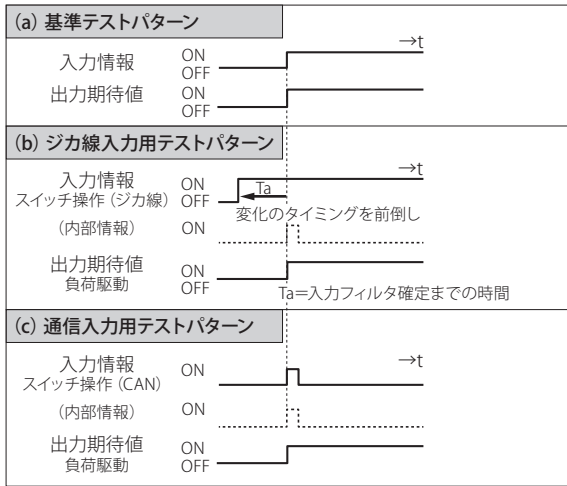


図3 信号入力タイミング変換の例

表1 信号割付設定パラメータの設定例

(a) 信号割付設定パラメータの説明

No	意味	説明
①	I/O種別	I: ECUへの入力 / O: ECUからの出力
②	実装種別	O: 実装 / N: 非実装
③	基本信号名	基準テストパターンで定義する信号名
④	対象Unit 信号名	変換先の信号名
⑤	対象Unit フレーム名	変換先のフレーム名 (CAN信号の場合)
⑥	信号種別	P: Port 信号 / C: CAN 信号
⑦	ON側調整時間 (ms)	OFFからONに変化時のタイミング調整時間
⑧	OFF側調整時間 (ms)	ONからOFFに変化時のタイミング調整時間

(b) 信号割付設定パラメータの設定例

①	②	③	④	⑤	⑥	⑦	⑧
I	O	SW_00	SW_X		P	30	30
I	O	CAN_Sw_00	CAN_Sw_X	F_CAN_Z	C	0	0

設定できるようにし、一つの設定ファイルのみで一元管理できるように工夫した。なお、タイミングを調整するための時間幅は、変換元となるテストパターンに対して相対的な時間として設定できるようにした。これにより、既に特定の車種用に作成済みのテストパターンがある場合には、改めて「基準テストパターン」を作成する必要はなく、別の入力タイプのテストパターンに直接変換することも可能になった。

2-3 割付変換ツールの適用と効果 入力信号のタイプが変更になり、変更後の車種用のテストパターンを作成するケースを想定して、①従来どおりの手作業でテストパターンを変更した場合、②今回作成した割付変換ツールを適用した場合、の作業工数を比較検証した。テスト対象は、流用元の車種から、9個の信号の入力タイプが変更になり、

その変更により該当の入力信号を含む、全体の2割の機能が影響を受ける後継車種に流用する場合を想定した。

検証により、①の手作業実施時に比べて②のツールを適用すると、テストパターンの変換工数を約20%削減できるという結果が得られた。また、ツールで自動変換することにより、対象信号の見落としによる変換の抜けモレを防ぐことができ、より早い段階でテスト設計品質を高めることができた。

3. モデルベース開発におけるテストの高効率化

3-1 モデルベース開発時のソフトウェアテストの課題

モデルベース開発では、設計段階でシミュレーションにより制御の正しさを検証する。そのシミュレーション技術には、車載ソフトウェア開発の上流工程から順に、設計したモデルの検証 (MILS)、モデルから生成したソースコード (プログラム) の検証 (SILS^{*7})、ECUにプログラムを組み込んだ状態での検証 (HILS)、の種類がある (図1)。

各検証段階では、同一の入力信号の組み合わせに対し検証対象であるモデル、プログラム、ECU実機がすべて同一の動きとなることを確認する必要がある。

このためには、MILS、SILS、HILS間でテストパターンを共通化してテストすることが、品質確保のためにも検証作業の効率化のためにも重要である。ところが、設計シミュレーションの実施対象であるモデルとECUでは、検証対象の範囲が異なるため、テストパターンの共通化に課題があった。以下に、設計シミュレーションと実機において、検証範囲が異なる理由を説明する。

一般に、車載ソフトウェアは、入力処理部とアプリケーション部と出力処理部の各モジュールから構成される (図4 (a))。この構成では、入出力インタフェースに変更があった場合に該当するモジュールのみを差し替えれば、アプリケーション部の作りに影響を与えることがない

(a) ECUのソフトウェアの構成



(b) モデルの入出力インタフェース

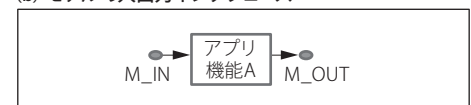


図4 ECUのソフトウェア構成とモデルインタフェース

ため、アプリケーションの流用効率が高まるというメリットがある。

このとき、MILSでは、このモジュール化された機能毎のアプリケーションの単位で設計シミュレーションを行う(図4(b))。そのため、MILSでは、ソフトウェアの内部信号を入力信号として使用するが、HILSでは、ECU装置のハードウェア外部の信号を入力信号として用いる。このように検証対象の外界とのインタフェース部が異なり、入力信号の確定タイミングの違いが生じる。

例えば、入力信号が通信インタフェースによって定期的に入力される信号である場合について、以下に説明する。

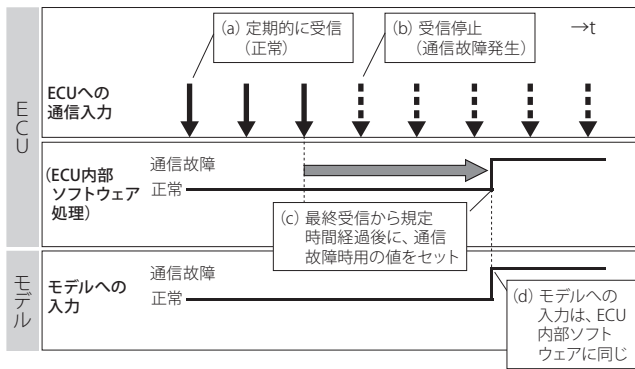


図5 通信故障時のECUとモデルの信号の差異

ECUを対象に通信故障のテストを行う場合、通信線に流れる信号を停止させる(図5の(a)から(b)への変化)。ECUに組み込まれたソフトウェアには、規定の時間が経過しても通信からの入力信号が得られない場合には、ソフトウェア内部の入力信号に通信故障時用の値をセットする処理が含まれている(図5の(c))。他方、設計シミュレーション用のテストシナリオでは、テスト対象のモデルは通信インタフェースと切り離された制御機能単独の単位なので、ソフトウェア内部の入力信号と同様の通信故障時用の値をセットし、入力する(図5の(d))。

そのため、MILS、HILSで同じタイミングで出力信号を出力させたい場合、HILS用のテストシナリオでは、車載ソフトウェア内部で通信故障が成立するよりも時間的に遡り、通信線に流れる信号を停止するパターンとする必要がある。

上記の理由により、モデル用のテストパターンはそのままECU対象のテストに流用することができなかつた。前章で課題となったのは、テスト対象の入力信号のタイプ違いであり、本章で課題となるのは、テスト対象がモデルとECUの違いの差はあるものの、何れも入力信号のタイミングの違いが問題であるので、先に述べた、流用開発時の入

力タイプ別のテストパターンの違いを吸収する手法が応用できると考えた。

3-2 HILS用入力信号変換プログラムの作成
ここで、このタイミングの違いを吸収するHILS用入力信号変換プログラム(以下、入力信号変換プログラム)を作成した。その変換処理の1例を図6に示す。

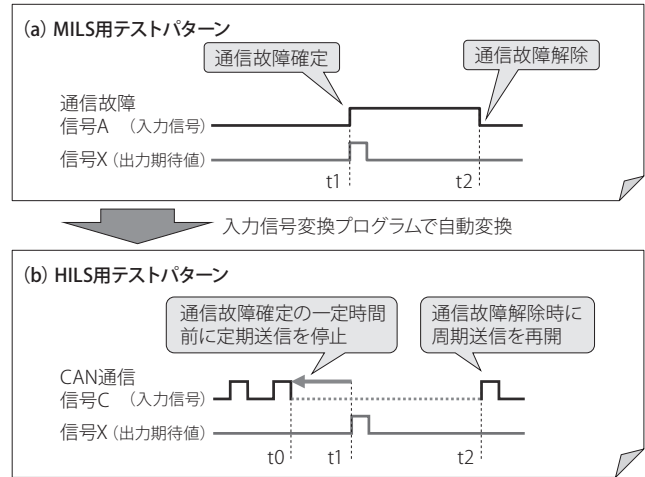


図6 通信故障時テストパターン変換の例

先に述べたように、MILS、HILSで同じタイミングで出力信号を出力させるためには、HILS用のテストパターンでは、ECU内部で通信故障が成立するよりも前のタイミングで、ECU外部での通信の故障が生じるよう設定する必要がある。

図6の例では、MILS用のテストパターンで通信故障が確定する時間 t_1 を基点に、通信故障判定の規定時間だけ遡った時間 t_0 に、通信信号Cの定期送信の停止を指示する波形を生成する。また、MILS用のテストパターンの通信故障解除時は、HILS用テストパターンに、同タイミング t_2 で定期送信再開の波形を生成する。

この入力信号変換プログラムには、2-2で説明した、入力信号の通信とジカ線の制御タイミングの違いを自動的に変換する機能も取り込み、MILS用テストパターンの内部的な信号変化を、HILS用にECUの信号入力タイミングへ変換できるようにした。

今回開発したテスト支援ツールの全体構成を図7に示す。本ツールは、MILS検証時とHILS検証時のテストパターンの差異を自動的に吸収する「入力信号変換プログラム」と「判定処理プログラム」から構成される。前者は、信号入力タイミングの違いを自動変換し、後者は、HILSの検証対象であるECUにハードウェアが含まれることによる出力誤差を自動的に吸収する役割を持つ。

これらのプログラムを自動テスト装置内に配置することで、あらかじめMILS用テストパターンをHILS用に変換したファイルを作成することなく、HILS検証中にリアルタイムに入力タイミングの変換を行う構成とした。これにより、入力タイミングが異なるだけのテストパターンファイルを複数保管する必要がなくなり、基本となるテストパターンファイルのみを管理対象とすればよく、ファイル管理の手間を効率化できた。

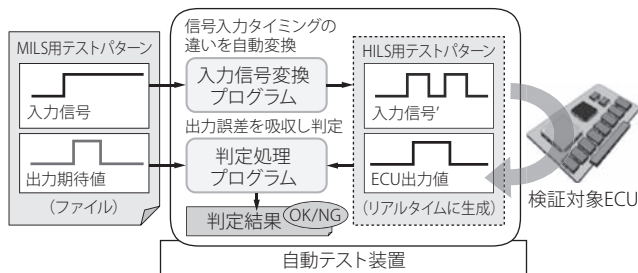


図7 テスト支援ツールの構成

3-3 開発プロセスへの適用と手順導入の効果 本ツールをモデルベース開発の手順に織り込み、実際の開発プロジェクトに対して適用評価を行った。評価対象としたのは、入力信号数が20個以上、出力信号数が8個の1機能である。

その結果、自動テスト装置の活用を前提に、テスト実施工数を従来比で約50%削減できるという検証結果が得られた。上流工程で設計シミュレーションを行い、そのテストパターンを製品テストにも応用してテストに活用することにより、製品テスト時の手戻りを防ぎ、従来より短時間で製品の品質を高めることができた。

4. 結 言

本稿では、高機能化や複雑化が進む車載ソフトウェアに対し、効率的なテストを行うための支援ツールの開発と、その導入効果について述べた。

第一の課題である車載ソフトウェアの流用開発時の品質確保の効率化については、テストパターンをECUの設計変更前と設計変更後で共通化を図る技術を構築し、該当するテストパターンの変換工数を20%削減できた。また、第二の課題に対するモデルベース開発の取り組みでは、前記のテストパターン共通化の技術を応用して、設計シミュレーション用のテストパターンを、ECU実機テスト用のテストパターンと共通化する技術を開発した。これにより、自動テスト装置の活用を前提に、テスト実施工数を従来より約50%削減し、また、製品テスト時の手戻りを防ぎ、より短時間で製品の品質を高めることができる目処を得た。

今後は、モデルベース開発プロセス下の評価技術のさらなる向上に努め、車載ソフトウェア開発のより一層の高効率化と高品質化に取り組んでいく。

用語集

※1 ECU

Electronic Control Unit：自動車に搭載される電子制御ユニット。

※2 MILS

Model in the Loop Simulation：設計したモデル段階のソフトウェアを検証するプロセス。

※3 HILS

Hardware in the Loop Simulation：実車両環境を模擬できるシミュレータを使用してECUを検証するプロセス。

※4 ジカ線

一つの信号のみを送信する導電ケーブル。

※5 CAN

Controller Area Network：電子回路や装置を接続するためのネットワーク規格。

※6 ボディ制御

ライトやドアロックなどの内装品などの電子制御。

※7 SILS

Software in the Loop Simulation：モデルから自動生成したソースコード段階のソフトウェアを検証するプロセス。

参 考 文 献

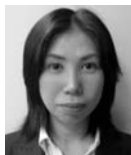
- (1) 金澤明 他、「車載ソフトの広範囲な起動タイミングの検証」、SEIテクニカルレビュー第172号、pp106-111 (2008)
- (2) 独立行政法人情報処理推進機構、「『組込みシステムの先端的モデルベース開発実態調査』調査報告書」(2012)
- (3) 堀川健一 他、「自動車ソフトウェアの標準仕様“AUTOSAR”の評価」、SEIテクニカルレビュー第175号、pp92-97 (2009)

執筆者

片岡 智美* : (株)オートネットワーク技術研究所
パワーネットワーク研究部 技師



坂 育子 : 住友電装(株) エレクトロニクス事業部



古戸 健 : (株)オートネットワーク技術研究所
パワーネットワーク研究部 グループ長



松本 達治 : (株)オートネットワーク技術研究所
パワーネットワーク研究部 部長



* 主執筆者