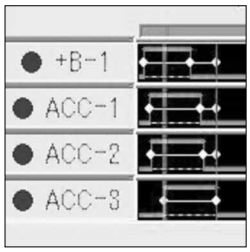


車載ソフトの広範囲な起動タイミングの検証

金 澤 明*・古 戸 健・松 本 達 治

"Power Switching Timing Test Tool" for Vehicle-Embedded Software Verification —— by Akira Kanazawa, Ken Furuto and Tatsuji Matsumoto —— To keep vehicle embedded software quality high, products must be verified on every event timing including unexpected timing.Especially important is the timing of power supply switching. In general, in the time region just after power supply is switched on, the possibility of occurrence of software error is higher than in the steady state. This is because the operating conditions at the time of switch-on of a power supply are different from those in the steady state, such as higher communication load.Continuous efforts are being made to verify product quality thoroughly, but if software scale and complexity continue to increase, the verification work might be extremely larger in the future than in these days. The authors have developed a testing tool that allows the verification to be performed at high time resolution at the timing of power supply switching. And the authors have also implemented automatic testing functions to the tool. Thus, they have constructed efficiency verification environment. The authors have applied this "power switching timing test tool" in the verification of prototype ECU. About 17,000 test cases were prepared and tested about prototype ECU. Such testing is impossible without the "power switching timing test tool". As a result, we have demonstrated the tool to be effective.In addition, we developed the automatic judge function to the tool so that testing operation efficiency is also improved considerably.

1. 緒 言

近年、自動車のドアロック、ルームランプ等の機能は使い易さと快適性が追求され、高級車だけでなく、小型車でも高機能化が進んでいる。これらの機能を制御する車載電子ユニット（以下、電子ユニットをECUと略す）も高機能化に伴って、搭載されるソフトウェアの規模は**図1**に示すようにこの20年間で約1,000倍に膨れ上がっている。そして、ソフトウェアの複雑化がこのまま進むと、今後の設計・検証不足による品質低下が危惧される。

一般的にソフトウェアの品質低下の防止策として、例えば以下の取り組みが考えられる。

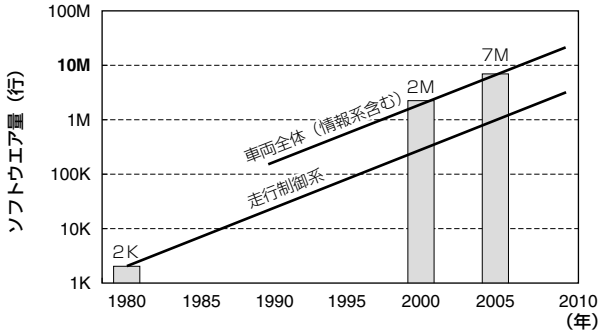


図1 車載ECUのソフトウェア量の遷移 (1)

① プロセスを改善してプロジェクトを管理レベルから改善する(2)
② 設計技法を改善して上流から品質を確保する
③ 開発経験豊富なメンバーによる技術的なレビューを実施する
④ 実機を使ってできるだけ多くの動作条件で検証する
⑤ シミュレーション環境で動作検証し、実機ではテストしにくいタイミングについて検証する(3)

本論文では、④の手法に着目し、実機を使って従来よりも格段にきめの細かい動作条件での検証を効率的に行えるツールを開発したので報告する。

2. 目 的

ソフトウェアの品質を確保するためには、設計段階で十分なレビューと検証を行い、設計ミスを完全に取り除くのが基本である。しかし、設計および設計の検証は人為的な作業であるため、ミスを100％除去できるとは限らない。そのため、ソフトウェアを実際に動作させてテストを行い、潜んでいる設計ミスを検出し、修正することで残存する設計ミスを無くしてから製品化する必要がある。

ソフトウェア開発における一般的なV字フローを**図2**に示す。このうち、ソフトウェアを実際に動作させて検証す

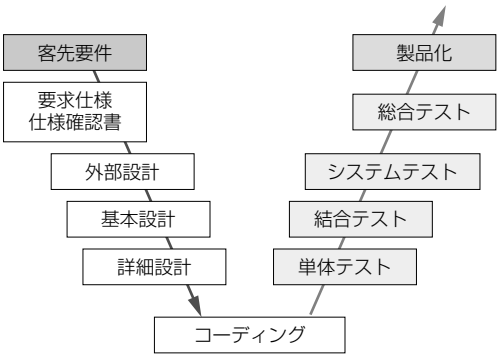


図2　ソフトウェア開発のV字フロー

るのは、システムテスト工程が中心である。今回はシステムテスト工程に着目し、車載ECUの起動タイミングの違いによる異常な動作がないことを効率的に検証することができるツールを開発したので、以下に報告する。

車載ECUに組み込まれているような組み込みソフトウェアは、スイッチ、センサ類、通信などの各種入力の発生タイミングの違いにより、動作条件が大きく異なる。ソフトウェアの品質を確保するためには、仕様書に記載された典型的な動作パターン、正常系の動作タイミングは言うに及ばず、想定外の特殊なタイミングまで含めて異常な動作がないことを広範囲に検証する必要がある。特に、ECUに電源を投入して初期化プログラムと通常処理が同時に動作し始めるタイミングは、通信負荷が増大するなど定常状態とは異なった動作条件となるため、過去の例から経験的にタイミング設計の抜けが潜みやすいことがわかっている。そのため、これまでは緒言に述べたソフトウェア品質を確保する個々の取り組みにより、製品の品質を確保してきた。しかし、今後益々ソフトウェアの大規模化と複雑化が進むと、従来通りの評価手法で品質は確保できるものの、検証作業にかかる工数が飛躍的に増大していく恐れがある。

そこで私は、パソコンを利用して車載ECUの電源起動タイミングを細かい時間分解能で少しずつ時間をずらして制御させて、その時の車載ECUの動作を効率的に検証できるツールを開発した。

3. 課　　題

まず従来から行われているシステムテストの手法とその課題について整理する。

電源の起動タイミングの違いによるテストを実施する場合は、何度も電源のON/OFF操作を繰り返して動作を検証する必要がある。

しかし、手作業で起動テストを行うため、以下の問題点がある。

① 100ms以下の精度で起動タイミングを狙ったテストが困難である。そのため、目標としたタイミングテストを実施するために何度も繰り返してテストしなければならない場合がある

② テストの結果を記録するのも手作業である。また、作業者への負担も大きくなるため、休憩や交代も考慮する必要がある。そのため、多大な工数が必要になり、開発コストが上昇する

これらは、いずれも手作業に頼っているために発生する課題であり、解決するためには、自動化ツールを導入する必要がある。

これまでにも、当社では市販の汎用的なテストツールを用いてテストを自動化してきた。例えば、複数イベントの同時発生など、設計上で問題が発生しやすそうなタイミングでのテストに取り組んできた。この市販のテストツールを応用することで電源のON/OFF動作を制御させることも検討したが、ツールの制約から、500μs以下の精度でのタイミング検証ができなかった。これまでは手作業による繰り返し起動テストで品質を確保してきたが、今後のソフトウェアの大規模化と複雑化により、品質を確保するために実施すべき繰り返し起動テストの回数が多くなることで、工数が飛躍的に増大していく恐れがある。

そこで、1μs精度で電源のON/OFFの制御を可能とし、電源起動のタイミングにおける動作を効率的かつ広範囲に検証するために、専用のツールを新規に開発することとした。

4. 原　　理

機器の構成を、**図3**に示す。また、仕様の概要を**表1**に示す。

今回開発したのは、配電／入出力BOXおよびパソコン上で動作する配電/入出力BOXの制御ソフトである。

外部からの直流電源が、配電/入出力BOXを介してECUに供給される。パソコンと配電/入出力BOXとは高速デジタル出力I/Fボード（図中高速DO）およびデジタル入出力I/Fボード（図中DIO）で接続されており、パソコン上で動作する制御ソフトから配電/入出力BOXにあるリレーのON/OFF状態を制御することで、ECUに供給している直流電源のON/OFFを制御する。

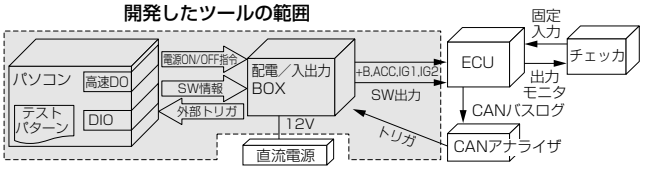


図3　機器構成図

表１　仕様概要

・独立して16種の電源制御が可能（4ECU群×4電源種（+B, ACC, IG1, IG2））
+B：バッテリ接続時、常にON状態である電源種
ACC, IG1, IG2：イグニッションキーの位置によりON状態となる電源種
・高精度なタイミング制御が可能
時間分解能：1μs
（但し、再現性能は配電／入出力BOXのリレーの性能に依存）
・１出力あたりの電流容量は40A
・テストパターンは汎用エディタで容易に作成可能
・外部にTTL出力が可能（ECUにポート入力可能）
・外部トリガによる連携スタート機能あり

パソコンからの電源信号出力に高速デジタル出力I/Fボードを採用することで、ソフトウェアの仕様としては1μs単位の高精度な時間分解能を実現している。ただし、再現テストを含めたツール全体の性能は配電/入出力BOXに実装するリレーの性能に依存する。リレーの応答性能に個体差がある場合はソフト的に補正することが可能であるが、ツール全体の応答性能を高めるためには高速なリレーを実装する必要がある。

また、配電／入出力BOXには、TTLレベルの信号を外部に出力する機能を備えさせることで、電源のON/OFF状態の制御と同様に信号の出力制御も可能とした。この機能によってスイッチのON/OFF状態をECU等に出力することもできる。例えば、スイッチと電源の状態が同時に変化した場合のテストなどに応用できる。

また、配電/入出力BOXに外部からテスト開始のトリガを入力することも可能である。制御ソフトは外部からのトリガを判断し、これと連携して電源やスイッチのON/OFF制御を開始することができる。

図３機器構成図に示した例では、ECUが出力するCANバス信号をバスデータ解析装置（図中CANアナライザ）に入力して特定のフレームを受信した際に配電/入出力BOXへトリガを出力している。ここで、CAN（Controller Area Networkの略称）とは、車載向けに開発されたシリアル通信プロトコルである。

このトリガにあわせて、ECUに対して車両電源系統である+B（バッテリ接続時、常にON状態である電源種）, ACC, IG1, IG2（イグニッションキーの位置によりON状態となる電源種）の4種類の電源のON/OFF状態を制御すると同時に、スイッチのON/OFF状態も制御している。

5. 利用方法

本ツール利用の手順は以下のとおりとなる。
① テストパターンを用意する。
② 用意したテストパターンをツールに入力、設定する。
③ ツールを使ってテストを実施する。

表２　テストパターンの記述フォーマット

コマンド	パラメータ	記述例
OutReq OutPut	ビット信号名称、相対出力時刻、出力状態	OutReq, ACC-1, 00000100ms, On
WaitTime	相対出力時刻	WaitTime, 15s
WaitTrig	ビット信号名称	WaitTrig, +B-1
Nop ExecOut	（なし）	ExecOut

テストパターンは、1行に1命令を記述する。記述内容は、コマンドの後にカンマ区切りでパラメータを記述する。テストパターンは汎用性の高いテキスト形式とした。**表２**にテストパターンの記述フォーマットを示す。

例えば、ACC電源のONになるタイミングが、+B電源のONになるタイミングの500ms後よりも遅れた場合のECUの挙動を検証する場合、**図４**のようなテストを実施する。テストパターンの例を**図５**に示す。この例ではACC電源のタイミングを1msずつ遅らせた場合のテストケースを示している。

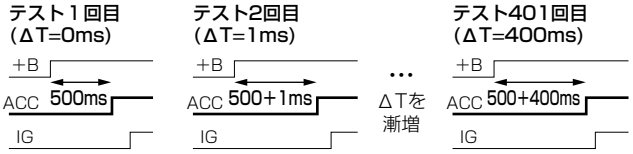


図４　ACC電源の立ち上がりが遅れた場合のテスト例

```
‘テスト1回目(Δt＝0ms)
OutReq, +B-1,                0ms, ON
OutReq, ACC-1,             500ms, ON
OutReq, IG-1,             1000ms, ON
ExecOut
                  :
WaitTime,                      5s
‘
‘テスト2回目（Δt＝1ms）
OutReq, +B-1,                0ms, ON
OutReq, ACC-1,             501ms, ON
OutReq, IG-1,              999ms, ON
ExecOut
                  :
                  :
WaitTime,                      5s
‘
‘テスト401回目（Δt＝400ms）
OutReq, +B-1,                0ms, ON
OutReq, ACC-1,             900ms, ON
OutReq, IG-1,              600ms, ON
ExecOut
                  :
```

図５　テストパターンの例

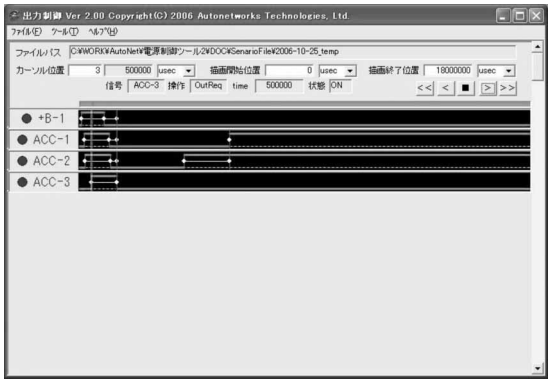


写真１　ツールの実行画面

写真２　実車ベンチへの接続例

テストパターンはテキスト形式で記述するため、汎用エディタで編集することができる。しかし、**図４**のように広範囲にタイミングを変化させたテストパターンを作成する場合、人手作業では効率が悪い。そこで、テストパターンを自動生成するツールを別途開発し、作業効率を向上させた。これについては紙面の都合により、詳細な説明を略す。

本ツールを起動し、作成したテストパターンを読み込むと、**写真１**のような画面になる。信号毎に、電源のON/OFF状態が波形で表示されるため、作成したテストパターンの電源ON/OFFタイミングが目的に適ったものかどうかが一目で分かるようになっている。

本ツールの応用的な使い方の1つとして、実車ベンチに接続することが考えられる。実車ベンチへ接続した例を、**写真２**に示す。

実車と同じ環境で、複数のECUに対して、様々なタイミングで電源のON/OFFを実施することができる。これにより、車両動作をイメージしながら、開発ECUの電源起動のタイミングに依存する動作を検証することができる。

6.　効果検証

実際に本ツールの有効性を検証するため試作ECUに適用して評価した。その構成を図8. に示す。試作ECUは、2つのCANバスの通信データを相互に中継する機能を有する装置である。

試作ECUの2つのCANバスには、他の車載ECUを模擬したECU（以下、模擬ECU）がそれぞれ2台ずつ計4台接続されている。制御パソコンから配電/入出力BOX経由で電源のON/OFFを制御するのは、試作ECUおよび試作ECUに接続された模擬ECU4台の計5台である。評価パソコン（CANバスデータ解析の機能あり）では、2つのCANバスのデータを収集してログに記録し、このログを解析することで試作ECUの動作を検証する。

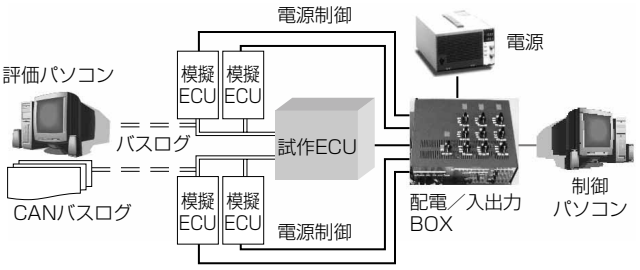


図６　試作ECUへの接続例

試作ECUに接続された2つのCANバスに、模擬ECUを2台ずつ接続し、試作ECUのテスト仕様をもとに、各模擬ECUの電源起動のタイミングを様々に変化させても正常に動作することを確認した。

具体的には、12通りの起動シーケンスをベースに、パターン違いも含めて合計16パターンを、1ms単位でタイミングをずらしたシナリオを合計11,020通り作成して評価した。また、仕様上のジャストタイミングを起点として前後20msの時間幅に限定した上で0.1ms単位で起動タイミングをずらしたシナリオも6,400通り作成して追加評価した。これら1ms単位でのテストシナリオ11,020通りと、0.1ms単位でのテストシナリオ6,400通りの計約17,000通りのテストシナリオを作成して、動作検証した。

実行結果は、2つのCANバスのデータを評価パソコンで収集してログに記録し、これを解析して仕様と比較することにより確認した。

ログの解析には、**図７**のようにログの比較ツールを開発して利用した。これは、あらかじめ用意したリファレンスとなるパターンデータ（設計意図を厳密に反映したデータ出力タイミングパターン）と実行結果（CANデータ）とを自動的に比較し、相違を求めることで設計した仕様との差異を検出するものである。

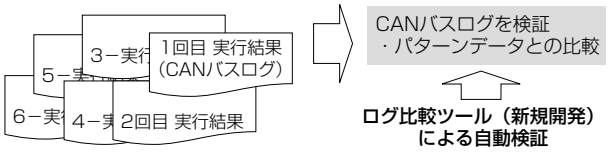
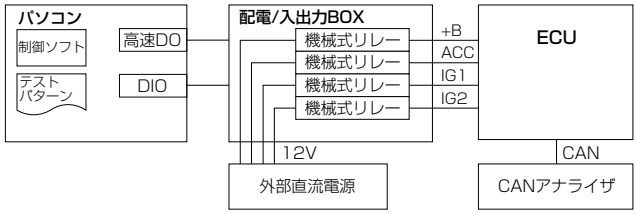
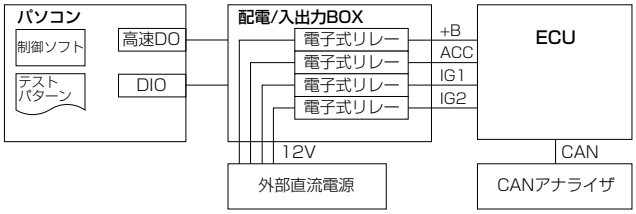


図7　ログ比較ツールによる自動検証



(a)　現在の構成



(b)　改善後の構成（例）

図8　配電／入出力BOXの現在の構成と改善後の構成（例）

7. 考　察

試作ECUの動作検証を対象として、約17,000通りの広範囲なテストパターンを作成し、電源起動のタイミングの動作検証を自動化することで効率化をはかった。

この作業を手作業で実施した場合に必要な工数は、次のように試算できる。電源スイッチの操作とテスト結果の確認・検証と記録を2名の作業者で分担するものとし、1テストパターンあたり30秒かかると仮定すると、17,000通りのテストパターンの実施には約140時間必要である。2名の作業者で実施しているため、工数としては約280人時かかる。

本ツールを使用した場合には、テストシナリオの作成に約4時間、テストの実行に約49時間かかり、結果の評価に約20時間かかった。このうち、テストの実行は本ツールによって自動的に行われるため、必要な工数は全部で24人時となる。

以上により、従来手法と比較すると256（280-24）人時、即ち32人日の工数削減が可能になる。

さらに、本ツールの発展的な適用用途として、以下の例が考えられる。

① 瞬断テスト

② バスショート

瞬断とは、ECUの電源が一瞬だけOFF状態になることを呼び、車両環境においては電線の接続不良などで発生し得る。瞬断発生時、ECUはあらかじめ定められた異常発生時処理を行う必要がある。この異常発生時処理が適切に行われているかどうかを検証する目的で、瞬断テストが行われる。

瞬断テストは、一瞬（例えば10μs）だけ電源をOFF状態にするテストパターンを作成すればよい。しかし、今回製作したツールは**図8（a）**のように配電／入出力BOX内のリレーとして機械式リレーを採用しているため、リレーの追従性が悪いと、一瞬だけ電源をOFFにしてもリレーが反応せずに瞬断が発生しない可能性がある。また、瞬断試験を繰り返しテストに使うと、リレーの接点が溶着を起こして装置が故障する可能性がある。そのため、機械式リレーは瞬断試験には適さない。

そこで、配電／入出力BOX内のリレーを、**図8（b）**のように電子式リレー（FET等）に置き換え、タイミングをより高精度とし、高速なON/OFF動作の繰り返しにも対応することで、瞬断テストなども可能となる。

そしてバスショートとは、通信線が電源ラインやグラウンド線に接触し、通信不能となる異常状態のことである。これについても、本装置を応用すればパソコンから定められたタイミングでバスショートを発生させることができ、動作条件の広範囲な検証に寄与できると考えられる。

8. 結　言

今回、これまでは500μsの精度でしか行えなかったタイミング検証を1μsの精度で可能とするソフトウェアと、そのベースとなるハードウェアを開発完了した。

試作ECUを用いた効果検証の際には、テストパターンの自動生成ツール、およびテスト結果の自動解析ツールを開発して利用することで、さらに効率化をはかり、ツールの有効性を示した。

今後の可能性として、瞬断やバスショートなどの異常系への対応が考えられる。今後も広範囲なテストツールを開発し続けることで、これまで以上に効率的かつ品質の高い車載ECU向けソフトウェアの開発に貢献していきたい。

参 考 文 献

（1）小川計介、「標準化で開発効率を高める車載ソフト巨大化に立ち向かう」、日経Automotive Technology 2007年11月号、pp82-97

（2）寺久保敏 他、「CMMレベル3に準拠した車載向けソフトウェア開発プロセスの構築」、SEIテクニカルレビュー第166号、pp45-50

（3）松本達治 他、「車載電子ユニット用ソフトウェアのクロス開発環境ツールの開発」、SEIテクニカルレビュー第165号、pp10-14

執 筆 者

金澤　明*：㈱オートネットワーク技術研究所　ソフト開発センター

古戸　健：㈱オートネットワーク技術研究所　ソフト開発センター　主任研究員

松本　達治：㈱オートネットワーク技術研究所　ソフト開発センター　センター長

＊主執筆者