

車載LANの ソフトウェア・テストを自動化する

——表計算ソフトを利用してLIN 2.0ドライバのテスト・プログラムを生成

名野 響

近年、自動車をはじめとする組み込みシステムの開発工程では、ソフトウェア開発の比重が高くなっている。中でも、コード量の増大や品質維持といった観点から、ソフトウェア・テスト工程におけるカバレッジや効率化が重要になっている。筆者らは車載ネットワークのソフトウェア・ドライバなどを開発しているが、テスト・プログラムの生成とテスト実行を自動化することで開発の効率を引き上げることを考えた。ここでは、筆者らが実際に開発したLIN 2.0 通信ドライバを例にとり、ソフトウェア・テストの自動化手法について解説する。(編集部)

組み込みソフトウェアに対する機能と品質の要求は、増大の一途をたどっています。その一方で、納期短縮という要求もあります。こうした要求にこたえるため、開発の多くの工程で作業の改善(あるいは改革)が必要となっています。とくにソフトウェア・テストの工程は品質確保の最後のとりです。要求仕様に対してもれなく効率的にテストを実施することが必要です(右掲のコラム「ソフトウェア・テスト自動化の現実」を参照)。

ここでは、LIN(local interconnect network)2.0のドラ

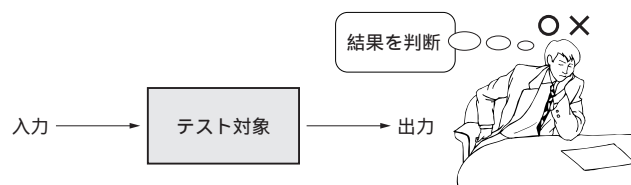


図1 テストの入力と出力

テストとは、テスト対象に与えた入力と得られた出力から結果を判断することである

イバ・ソフトウェアを例に、ソフトウェア・テストの効率化について考えます。現在、ほとんどの組み込みソフトウェアはなんらかのインターフェースによって他者(ハードウェアや上位アプリケーションなど)と通信しているはずです。今回取り上げるLIN 2.0は通信ソフトウェアですが、ほかのタイプのソフトウェアにも応用できると思います。

1 自動化できる項目を見定める

一般に、テスト対象に対して何かしらの「入力」を与え、それによって得られた「出力」を判別することで、テストの可否を決定します(図1)。今回のテストでは、この「入力」と「出力」を次のように定めています。

- 入力：テスト実行プログラム+テスト・パラメータ
- 出力：テスト結果、テスト番号、エラー・コード

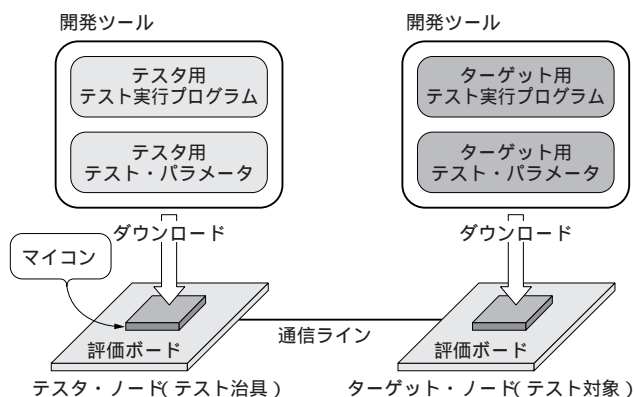


図2 テスト・システムの構成

テスト治具とテスト対象を通信ラインで接続し、テストを実施する

KeyWord

ソフトウェア・テスト, LIN 2.0, ドライバ, テスト・パラメータ, プログラム・ファイル, μ PD78F0148H, 78K0/KF1+, μ PD70F3239, V850ES/FJ2, PM+, MULTI 2000, ビット・エラー

column

ソフトウェア・テスト自動化の現実

「テストの自動化」。どの開発現場でも品質向上とコスト削減の切り札としてこのことを信奉しているはずですが、しかし、実際のところ「言うは易く、行は難し」といった状況ではないでしょうか。その原因は、大きく二つのカテゴリで考えられます(表A-1)。

1) 物理的に自動化しにくいもの

2) 担当者の意識の問題

さらに自動化する以前に、テスト項目をどのように(もれなく、効率良く)抽出すればよいかという問題があり、これが意外と悩みどころ

であると思います。本稿で紹介している事例は、要求仕様に対してもれなくテスト項目を抽出するという点に着眼しました。そして、自動化については市販のツールではなく手元にあるツール(今回はExcel)をくふうして使用したという点もポイントです。

さらに大きなシステムでは、「もれなく」の解決にまじめに取り組むと、天文学的な数のテスト項目を扱わなければならないという事態に陥ってしまい、これを「問題のないように」省く技術が必要になります。

表A-1
「自動化」を実現しにくい理由

カテゴリ	例
物理的に自動化しにくい	- 動作をパターン化しづらいシステム - 入力を自動化しにくいシステム - 出力を読み取りにくい(人が読むしかない場合) - 良不良の判断が単純にできない入出力関係
担当者の意識の問題	「自動化」と言われても、何を行えばよいかわからない - 自動化する環境を作るくらいなら手でやったほうが手っ取り早いと信じている - 自動化にはそれなりのツールを買い取らなければならないと信じている

テストを実施するシステムの構成を図2に示します。テスト治具(ここではテスト・ノードと呼ぶ)とテスト対象(ここではターゲット・ノードと呼ぶ)を通信ラインで接続します。そして、開発ツールを用いてテスト・パラメータとテスト実行プログラムをそれぞれのノードにダウンロード(入力)し、テストを実施します。

ここでテスト・パラメータとは、テストを実施するテスト治具やテスト対象(以降、これらを合わせて「テスト・システム」と呼ぶ)のふるまいを定義した値です(詳細は後述)。テスト・ノードとターゲット・ノードでテストを実行するプログラムを共通にし、要求仕様に対応したふるまいごと

のテスト項目をパラメータ化することで自動化を目指しています。

テスト結果はツール上で変数の値を直接確認するか、またはログ・ファイルを参照して確認します。ほとんどの項目はログ・ファイルに記録でき、期待値との一致検証を行うことで完全に自動化できます。

今回のテストの流れを以下に示します(図3)。

- 1) テスト項目の抽出——テスト対象となる通信ドライバの仕様書からテスト項目を抽出し、テスト仕様書を作成する
- 2) テスト・パラメータの作成——抽出した各テスト項目のテスト・パラメータを作成する

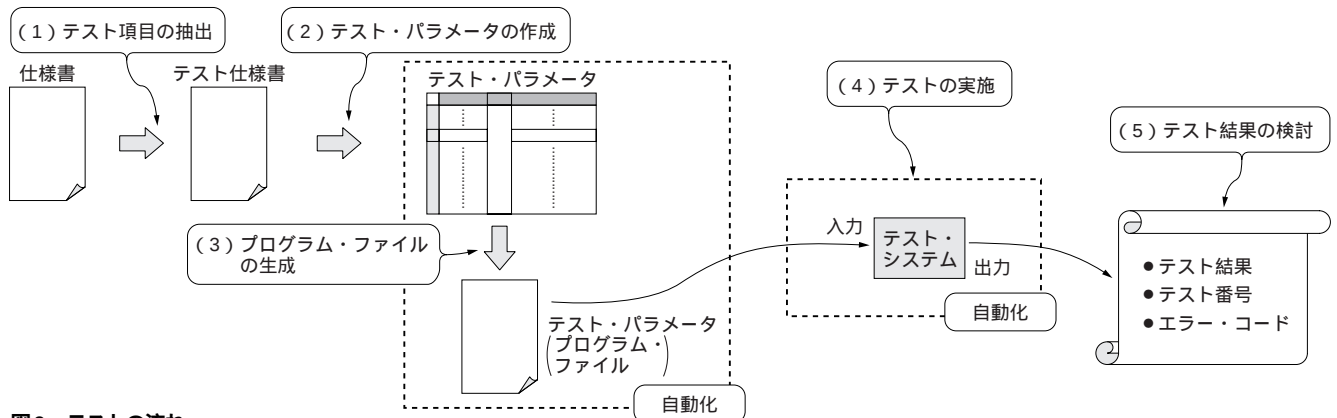


図3 テストの流れ

要求仕様からテスト項目をもれなく抽出する。その後、抽出したテスト項目からテストのふるまいを定義するテスト・パラメータを作成する。プログラム・ファイルの生成からテストの実行、テスト結果の出力までは自動で行う