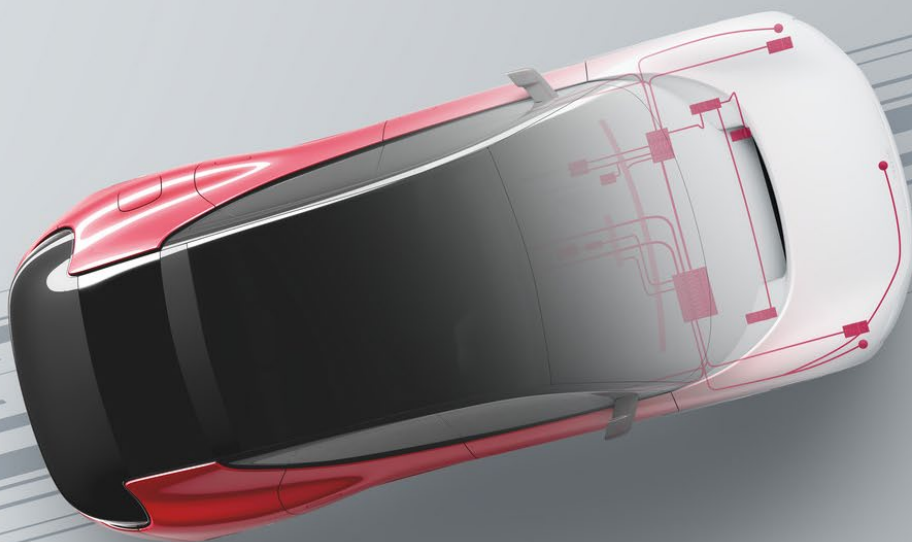


VECTOR >

はじめての vTESTstudio

ビギナー向け資料「はじめてシリーズ」



目次

はじめに.....	4
用語集	5
1. ベクターテストソリューション概要と vTESTstudio の役割.....	7
1.1 CANoe 概要	7
1.1.1 CANoe シミュレーション機能(IL 機能)	8
1.1.2 CANoe テスト機能	10
1.2 vTESTstudio の役割	13
1.3 VT システムの役割.....	18
2. どんなことができるのか?	21
2.1 vTESTstudio で容易にできることの例	21
2.2 Test Table Editor の概要	24
2.3 CAN 通信のチェック	28
2.4 診断通信のチェック	31
2.5 刺激に対する応答を確認	33
2.6 作成済みのテストの再利用	38
2.6.1 テストケースを再利用する.....	38
2.6.2 Parameter Editor の使用.....	43
2.6.3 テストケースのパーツを再利用する.....	46
2.6.4 テスト対象のシグナルやメッセージの汎用化	49
2.7 固有の仕様に基づく複雑なテストを作成.....	51
2.8 波形を定義して ECU への刺激として入力	53
2.9 テスト全体を一覧表示し、テストの抜け漏れチェック	55
2.10 その他の主な機能	58
3. よくあるお問い合わせ	60
3.1 vTESTstudio の操作方法を詳しく知りたい	60
3.2 CANoe など vTESTstudio 以外のツールの使い方が知りたい	60
3.3 CANoe でもテスト実装できますが、vTESTstudio を使用するメリットは何ですか?	61
3.4 実際に導入した事例について知りたい.....	61
3.5 テスト自動化をどこから、どのように進めたらよいのか分かりません.....	61
3.6 CAN 以外のプロトコルについてもテスト実装できますか?	61
3.7 ダイアグラムエディターはオプションですか?	62
3.8 vTESTstudio の実装内容について vTESTstudio を持っていない人にレビューしたい.....	62
3.9 ECU 内部値のテストはできますか?	62
3.10 VT システムと vTESTstudio との連携はできますか?	62

3.11 Simulink などのモデルに対してテストを行うことはできますか？ 62

3.12 実装中の ECU ソフトウェアに対してテストを行うことはできますか？ 62

あとがき 63

4. お問い合わせ窓口 64

発行元:ベクター・ジャパン株式会社

ベクター・ジャパン株式会社の書面による許可なしに、本書の内容を転載、複製、複写することを禁じます。
記述されている内容や製品画面の表示名称は予告なく変更されることがあります。

はじめに

本稿は、ベクターが提供する ECU 自動テストの設計・実装ツール「vTESTstudio」について、ECU テストの自動化を検討されている方や、vTESTstudio がどのようなツールか詳しくご存知ない方など、vTESTstudio のご使用経験のない方を対象に作成した資料です。vTESTstudio を使って「実際にどのようなことができるのか」、「どんなメリットがあるのか」など、vTESTstudio に関する基礎知識とユースケースについて、やさしく解説していきます。また vTESTstudio をすでにお持ちの方についても、使ったことのない機能や、より効果的な使い方などを知る機会となるよう、有益な情報を盛り込んでいますので、お読みいただければ幸いです。

本稿では CAN 通信を前提として説明しますが、他のプロトコル(LIN や車載 Ethernet など)をご使用の場合にも参考にいただけます。

上述のとおり、本稿は vTESTstudio でどのようなことができるのか?といった概要について説明する資料となります。そのため詳しい操作方法などは解説していません。vTESTstudio を実際にお使いになる場合には、以下を参考にしてください。

- > vTESTstudio トレーニング
- > vTESTstudio に関するアドバンストマニュアル
- > vTESTstudio の自習用サンプル(「CANoe テスト機能、vTESTstudio 自習用サンプル」)

トレーニングはベクターにて開催しており、以下から内容や開催日程を確認できます。

https://vector-academy.com/vj_training_jp.html

アドバンストマニュアルや自習用サンプルは、ベクター製品ユーザー向けコンテンツを提供している「ベクター・ジャパン Member Area」に掲載しています。

詳細は下記リンク先をご覧ください。(ベクターの Web サイトへ移動します)

<https://www.vector.com/jp/ja/support-downloads/vj-memberarea/>

それでは、まずベクターの ECU テストソリューションの概要と、その中での vTESTstudio の役割から見ていくことにしましょう。

用語集

本稿で記載している弊社製品に関する用語のうち、主なものを以下の表にまとめました。ただ、必ずしもすべて読む必要はありません。本文を読み進めていてわからない用語が出てきた際に、ここに用語集があったことを思い出し、参照いただければと思います。

用語	内容	初出
vTESTstudio	「ブイテストスタジオ」と読みます。本稿で説明する、ベクターのテスト設計・実装ツールです。	はじめに
CANoe	「キャヌー」と読みます。車載ネットワーク開発における総合的なツールで、ベクターテストソリューションの中核となる製品です。本稿では主に、テストの自動実行に関する機能を紹介します。	1章冒頭
VTシステム	「ブイティーシステム」と読みます。ECUの統合テストを実施する際に必要となる、電圧値などの電気信号の測定や刺激を行うためのハードウェアとなります。	1章冒頭
CANalyzer	「キャナライザー」と読みます。CANoeの機能のうち、「解析」と一部のシミュレーション機能を取り出した製品となります。	1.1節
IL機能	Interaction Layer、インタラクションレイヤーの略で、通常「アイエル」と読みます。CANoeのシミュレーション機能の1つで、データベースに基づきメッセージを自動で送受信します。	1.1.1項
Test Report Viewer	CANoeに付属するレポート解析用のツールです。CANoeが出力するテストレポートの閲覧・解析を容易に行うことができます。	1.1.2項
テストレポート	CANoeでテスト実行後、自動生成されるファイルで、テストの判定結果や詳細な情報が記載されています。本稿ではテストレポート、または単にレポートと記載します。	1.1節
Test Table Editor	vTESTstudioの持つさまざまなエディターの中で、最も標準的で、最も使用されるエディターです。本稿 2.2節にて概要を記載し、以降に続く2章の各節にてさまざまなユースケースを紹介しています。	2.1節

用語	内容	初出
コマンド	vTESTstudioでは、ECUのテストにおいて頻繁に使用される処理をまとめて定義済みコマンドとし、このコマンドを使用するだけで容易にテスト実装できる形となっています。本稿ではコマンドやテストコマンド、定義済みコマンドと記載しています。	2.1節
テストケース	vTESTstudioではテストのひとかたまりをテストケースと呼びます。一般用語の「テストケース」と、基本的に意味は同じです。	2.2節
値テーブル	値の意味が分かりやすいようラベルが貼られたデータで、実態は数値です。CANoeやvTESTstudioでは値の代わりにこのラベルを表示させることができます。	2.5節
Testcase Definition	Test Table Editor上で定義する、テストケースのひな型です。作成方法は通常のテストケースと同様ですが、引数を受け取ることができ、テストケースを汎用化することができます。	2.6.1項
テストケースリスト	Test Case Definitionをリスト形式で一度に大量に呼び出すことができる、vTESTstudioの効率化機能です。	2.6.1項
Parameter Editor	vTESTstudioの持つ補助的なエディターで、テスト用の定数を定義するパラメーターファイルを作成・編集します。	2.6.2項
パラメーターファイル	テストで使うさまざまな値を一元管理できるファイルです。テストケースから値を分離することで、後から値の変更や追加が必要になった際の工数を削減できます。通常の定数を定義するだけでなく、リストや構造体、構造体リストの定義も可能です。	2.6.2項
Function Definition	Test Table Editor上で定義する、テストケース内の処理のひな型です。よく使用する一連の処理をまとめ、分かりやすい名前を付けることで、容易に処理を再利用できるようになります。	2.6.3項

表 A: 本稿で記載しているベクター製品に関する用語をまとめた用語集。最初に読む必要はありません

1. ベクターテストソリューション概要と vTESTstudio の役割

ベクターでは ECU 開発に関わるさまざまな製品やサービスを提供しており、ECU テストに関する製品群もその 1 つです。ECU テストに関する製品のうち、ECU のシステムテスト用に広くご使用いただいている製品が、CANoe、vTESTstudio、V T システムとなります。この 3 つの製品はそれぞれ独立して用いられるわけではなく、CANoe を中心として、残りの 2 つの製品が CANoe の機能を補う形で使用されます。

本章では、これら 3 つの製品の概要とそれぞれの役割について解説していきます。

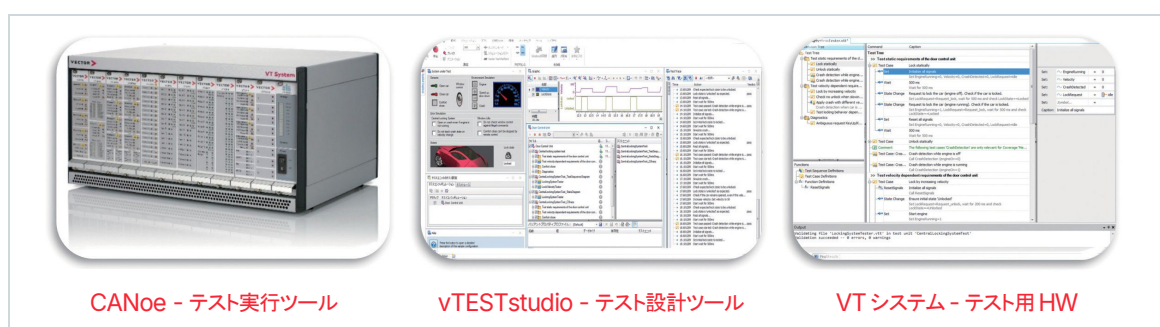


図 1: ベクターテストソリューションのなかで広く利用される 3 つの製品

1.1 CANoe 概要

CANoe は車載ネットワーク開発に関する総合的なツールで、CAN、LIN、車載 Ethernet などの車載ネットワークに関し、大きく分けて「解析」、「シミュレーション」、「テスト」の 3 つの機能を持っています。なお余談ですが、ベクターには CANalyzer という製品もあり、こちらは CANoe の上記 3 つの機能から解析機能(と一部のシミュレーション機能)を取り出したものとなります。

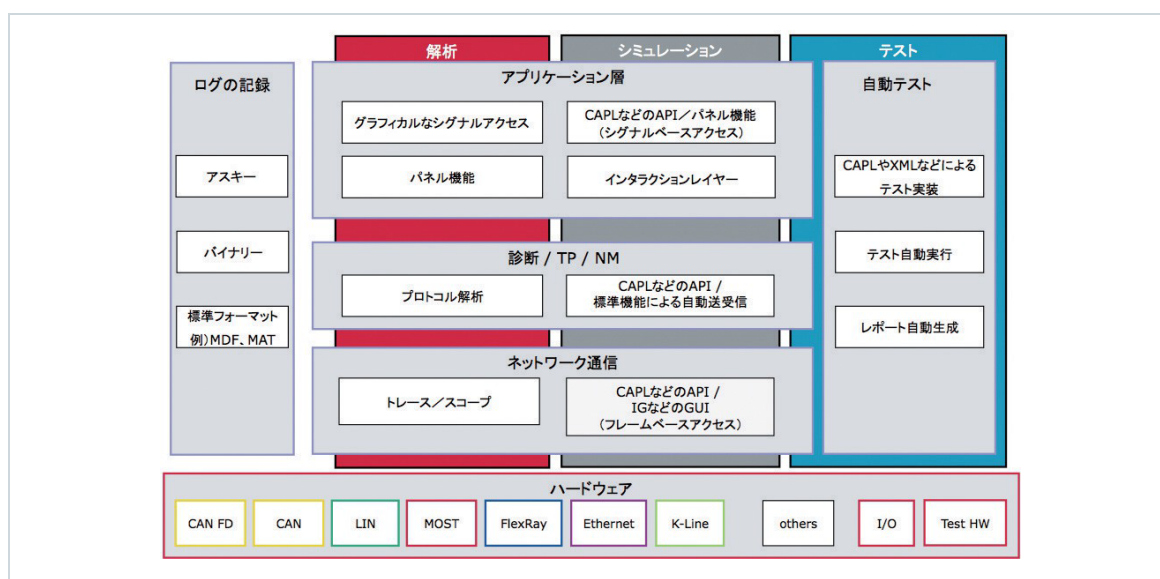


図 2: CANoe は車載ネットワークに関し、大きく分けて「解析」「シミュレーション」「テスト」の 3 つの機能を持つ

CANoe のテスト機能は極端に言うところ「テスト実行」を行う機能となります。事前に定義されたテストシナリオを自動実行し、テストの判定結果を記載したテストレポートを自動で出力します。vTESTstudio はこのテストシナリオを設計・実装するツールとなります。

CANoe のテスト機能は多くの場合、シミュレーション機能をベースとして利用しています。そのためまずは CANoe の「シミュレーション」機能について解説します。

※本稿では CANoe のテスト機能と、その前提となるシミュレーション機能に関する概要のみ記載します。その他の CANoe の機能につきましては、「はじめての CANoe」や、CANoe に関するアドバンスマニュアルを参照してください。

1.1.1 CANoe シミュレーション機能(IL 機能)

CANoe のシミュレーション機能とは、簡単にいうと ECU の通信相手をする機能です。テスト対象 ECU が車載ネットワーク通信を行う際には多くの通信相手が必要ですが、この通信相手を模擬する機能が CANoe のシミュレーション機能となります。CANoe は以下の図 3 のように、内部にシミュレーションノードを複数作成することができ、ネットワークに参加する複数の ECU を同時に模擬することができます。

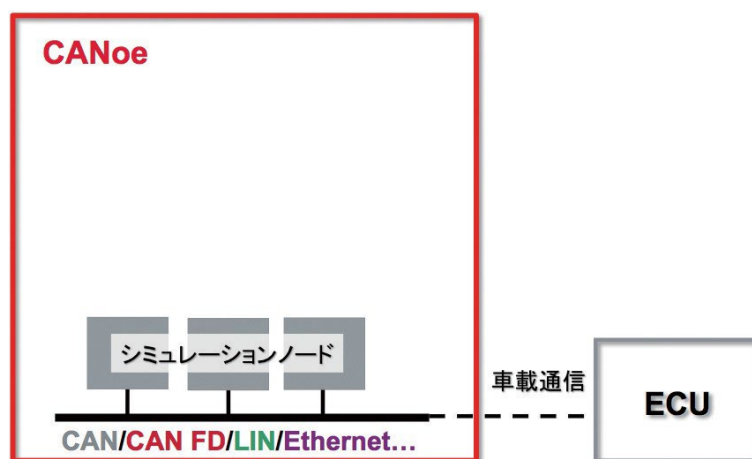


図 3: 3 つの製品の位置づけ、CANoe シミュレーション機能のみ記載。ECU の通信相手となるシミュレーションノードを複数作成可能

ただ、通信相手となる ECU の振る舞いを模擬するには、CANoe 上でそれぞれの ECU の振る舞いを実装しなければなりません。CANoe には CAPL(読み方:キャプル)という専用のプログラミング言語があり、この言語を用いてそれぞれのシミュレーションノードの振る舞いをプログラミングできますが、ECU の振る舞いすべてを実装するのは大きな工数がかかります。CANoe は、この課題に対応するための機能をいくつも取り揃えており、その一つに IL 機能があります。

■ CANoe IL 機能

ECU が他の ECU と通信を行う際には、あらかじめ定義されている通信データベースに基づいてメッセージの送受信を行います。たとえば CAN 通信の場合は送信するメッセージの ID や送信データの内容、送信する周期やイベント送信の方法などがデータベースに定義されています。CANoe で ECU の振る舞いを模擬する際にも、実 ECU と同様に通信データベースに基づいたメッセージの送受信を行う必要がありますが、CANoe には IL 機能があり、IL がデータベースの内容を読み取り、自動的にメッセージの送受信を実行します。これにより、シミュレーションノードの実装工数を大幅に削減できます。

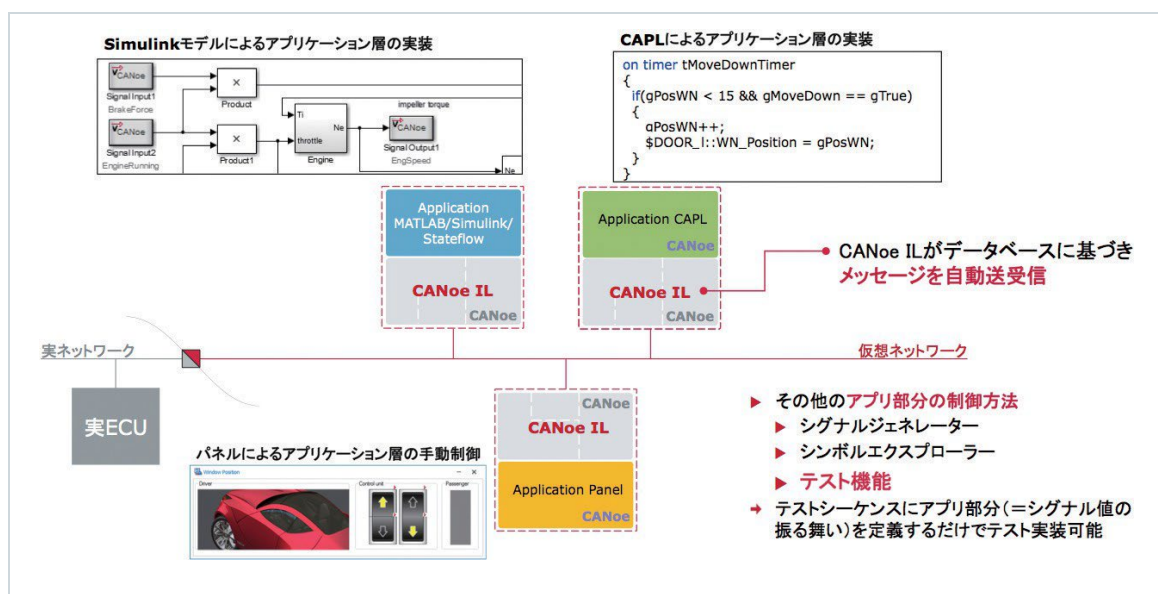


図 4: IL はシミュレーションノードの一部であり、データベースに基づきメッセージの自動送受信を実行

CANoe IL がメッセージの送受信を自動で実行するため、メッセージに紐づけされたシグナル(=制御データ)の振る舞いさえ実装すれば、通信相手の模擬ができるようになります。図 4 にあるとおり、このシグナル値の振る舞いを実装する方法として、前述の CAPL の他に Simulink モデルを利用する方法や、GUI で簡易的に手動操作する方法、そしてテスト機能を利用する方法があります。これは逆の見方をすると、CANoe テスト機能は CANoe IL をベースとして利用している、と言い換えることができます。

なお、CANoe シミュレーション機能にはこの他に CANoe コンフィギュレーションの自動生成機能や OEM パッケージなどさまざまな効率化機能がありますが、本稿では割愛します。

1.1.2 CANoe テスト機能

一般的に、テスト自動化ツールを用いて自動テストを作成し実行させるためには、コンピューターがテストのシナリオを読み込んで実行できるようにプログラミングする必要があります。CANoe でもこれは同様ですが、前述のとおり CANoe には IL 機能があるため、通信データベースに基づくメッセージの送受信については実装する必要がありません。テストシナリオ作成時には複雑なメッセージのやり取りについては気にせず、シグナル値の振る舞いだけを考慮して実装すればよいことになります。

IL 機能を用いることで CANoe と ECU との通信は成立しており、テスト機能はこの環境をベースとして成り立つことから、1.1.1 項で用いた図 3 に対してテスト機能を追加すると図 5 のように表すことができます。

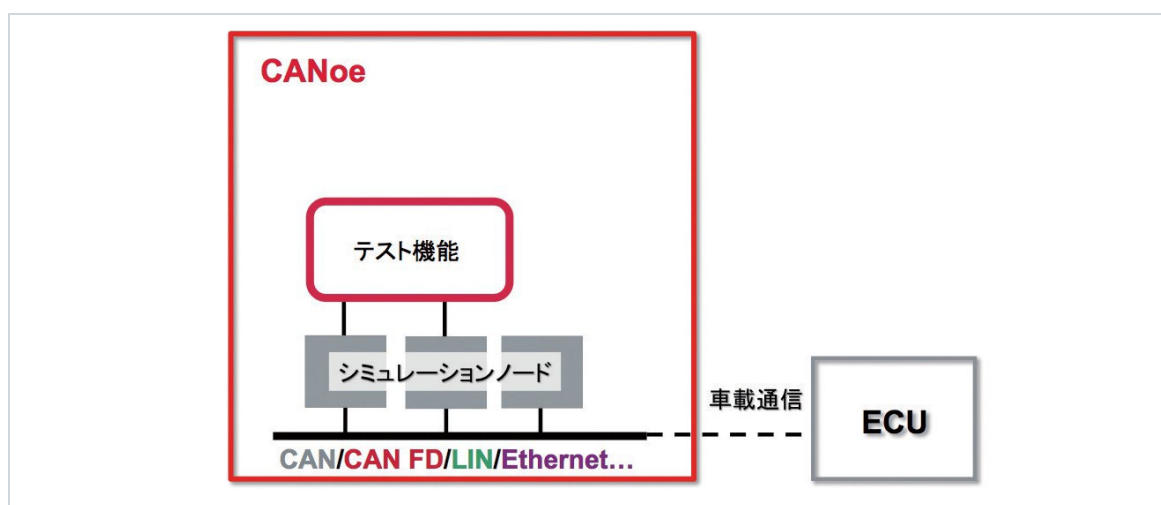


図 5: 3 つの製品の位置づけ、CANoe テスト機能を追加。テスト機能はシミュレーション機能をベースとして利用している

■ 自動テストのワークフロー

自動テストの実行には、まずテストを実装し、次にテストを実行し、最後にその結果を分析することになります。実際には更にこの 3 つのワークフローを管理ツールや CI ツールなどで管理しながら複数回サイクルを回すこともありますが、ここでは単純に上記 3 つのフローについてのみ考えます。

手動でテストを実施する場合、テスト実装はテスト仕様書に基づきチェックリストを作成するなどの方法で実施されることが多く、極端なケースではテスト実行者の頭の中に存在するだけのこともあります。テスト実行は手作業で行い、結果の分析は目視で行うことになります。

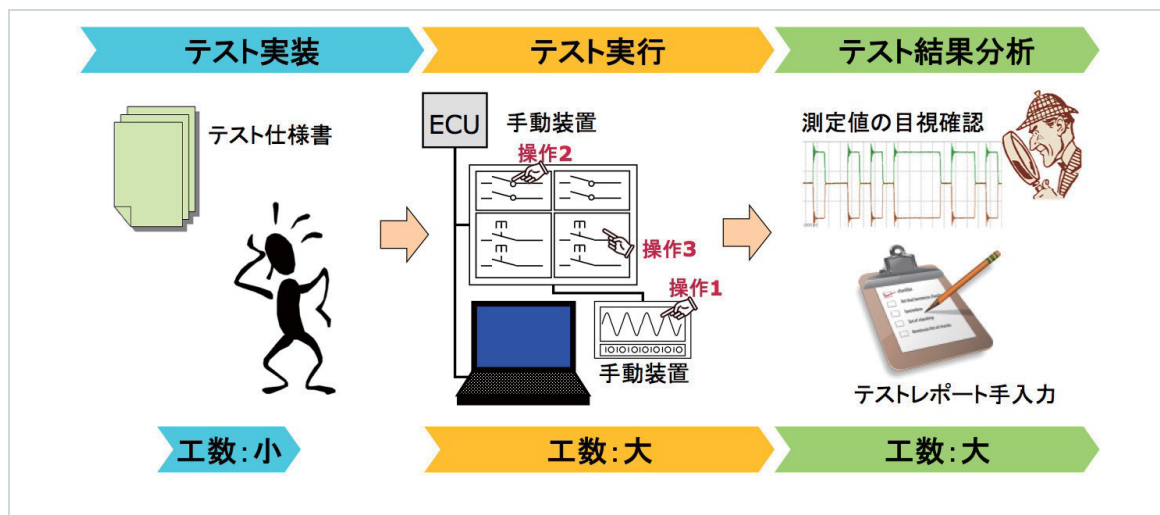


図 6-1: 手動でテストをする場合のフローと工数のイメージ。テスト実行と判定、レポート作成の工数が大きい

一方、CANoe を用いてテストを自動化した場合には、以下のようにテスト実装以外は自動化されます。

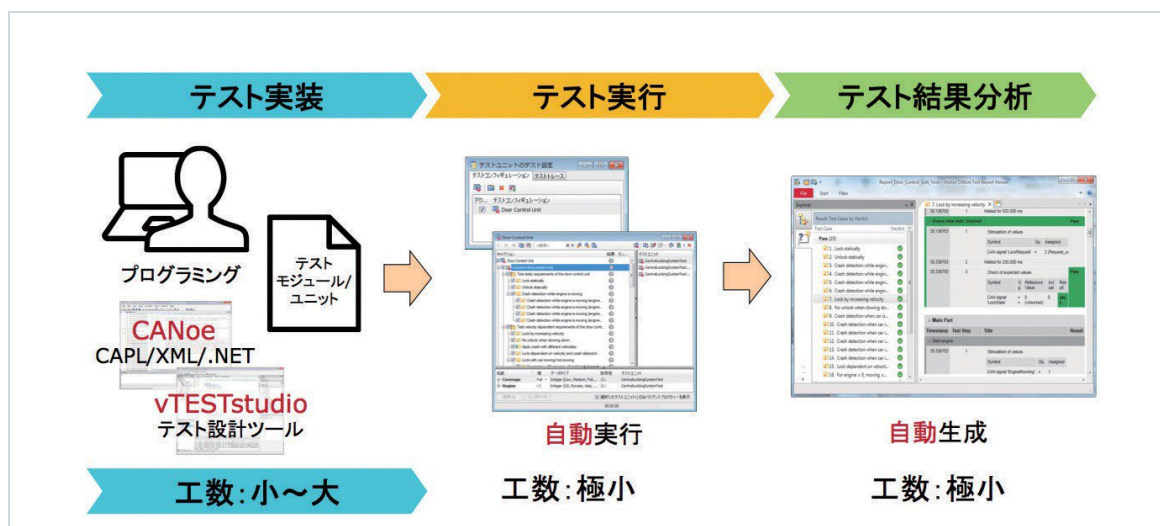


図 6-2: 自動でテストをする場合のフローと工数のイメージ。テスト実行と判定、レポート作成は自動化されるが、テスト実装工数は増大

自動化されるテスト実行とレポート生成の工程は、実際の CANoe 上の画面では以下のように行われます。

CANoe に読み込んだテストのシナリオに対し、テスト実行ボタンをクリックすると CANoe が自動的にテストを実行します。テストが終了すると、自動でテストレポートが生成されます。レポートボタンをクリックすると Test Report Viewer という CANoe に付属のレポート解析用のツールが起動し、結果の解析を容易に行うことができます。

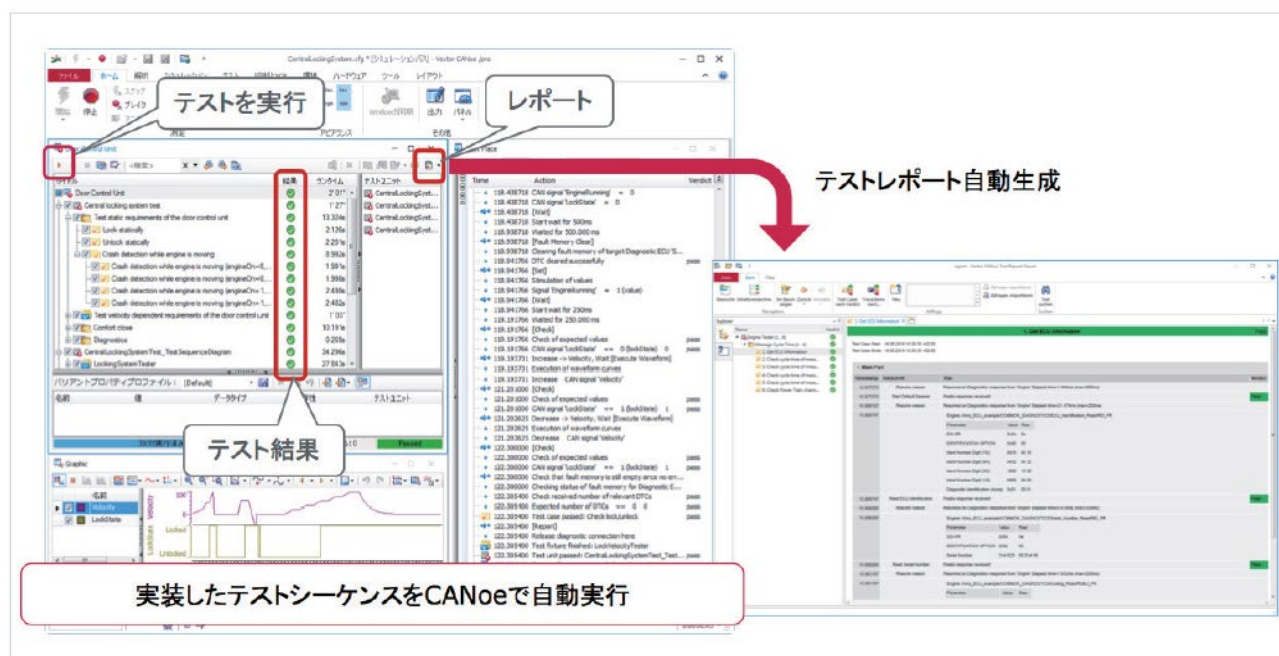


図 7: CANoe でのテスト自動実行とレポート自動生成

一方、テスト実装は手動でテストしていた際には工数としてはあまり大きくありませんでしたが、自動テストでは工数のかかる唯一の工程となった上、テストのシナリオをプログラミングするため、必要となる工数も大きく増加します。したがって、テスト自動化を進めるためには、テスト実装にかかる工数をいかに削減できるかが重要となります。

1.2 vTESTstudio の役割

CANoe には前述のとおり CAPL という専用のプログラミング言語があり、テストのシナリオもこの CAPL で実装することができます。また CAPL 以外にも、XML 形式の記述や.NET のプログラミング言語(C#など)を用いてテスト実装する手段も提供しています。更に、テスト専用の CAPL 関数など、テストに特化した API が準備されているため、通常のプログラミング言語を使用するよりも少ない工数でテスト実装できるようになっています。

しかし、プログラミングが必要であることは変わっておらず、テスト実装に必要とされる工数やスキルレベルは依然として高いままとなっています。この問題を解決するために作成されたのが、vTESTstudio となります。

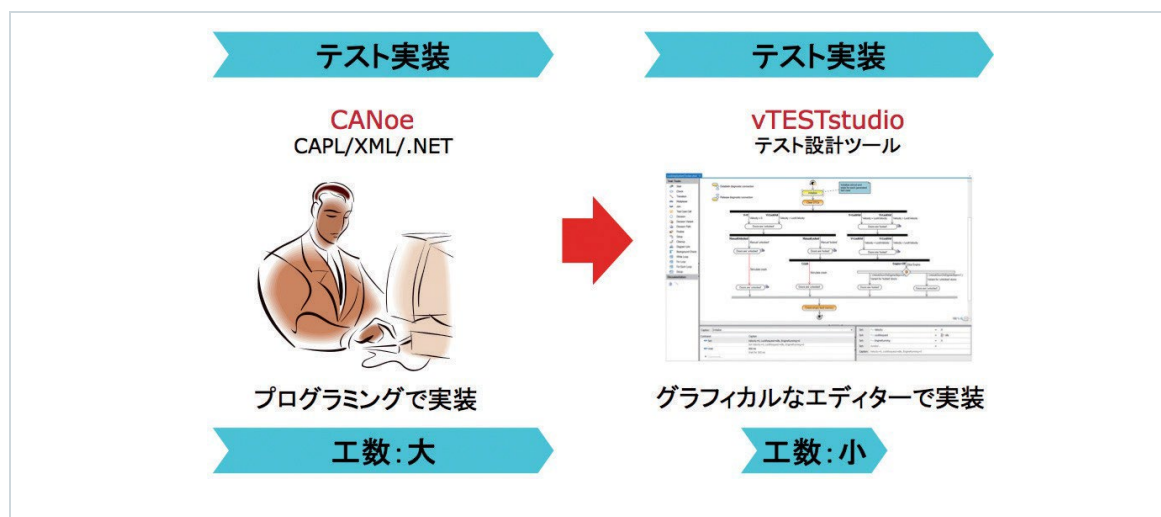


図 8: vTESTstudio の役割 = テスト実装を効率化

vTESTstudio は CANoe で実行するテストのシナリオを設計・実装するための専用ツールで、テスト実装を効率的に行うための機能を幅広く提供しています。vTESTstudio が無くとも、テスト実装自体は前述のとおり CANoe 単体でも行うことができます。しかしテスト自動化にはテスト実装の効率化が最も重要なポイントであり、vTESTstudio の提供するさまざまな効率化機能を用いることで、テスト自動化を容易に進めることができます。

ここまでで説明した CANoe 単体でのテスト実装と Test Report Viewer、vTESTstudio を前述の図 5 に追加すると、図 9 のように表すことができます。図中では、CANoe 単体でテスト実装を行った場合と、vTESTstudio を用いてテスト実装した場合の双方を記載していますが、どちらか片方で実装すればテストを実行できます。

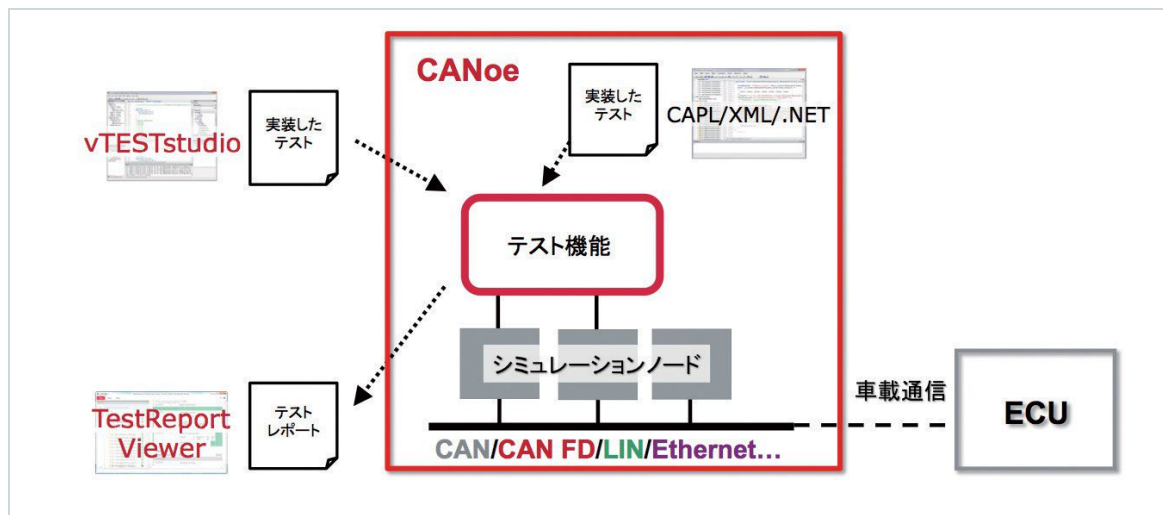


図 9: 3 つの製品の位置づけ、vTESTstudio などを追加。vTESTstudio で実装したテストを CANoe でテスト実行

■ CANoe と vTESTstudio の連携

vTESTstudio は CANoe とは別の独立したアプリケーションです。しかし、vTESTstudio は CANoe 専用のテスト実装ツールですので、双方向の連携が容易に行えます。

CANoe と vTESTstudio の間で行われるやり取りには、図 10 の中央に示したように、2 種類あります。

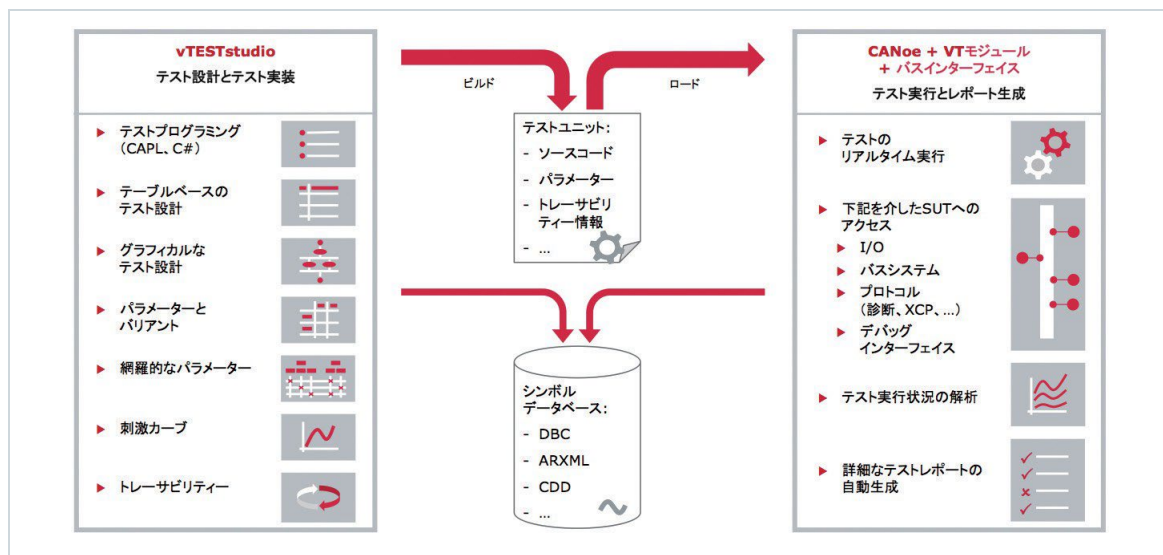


図 10: vTESTstudio と CANoe との間で行われる 2 種類のやり取り
 上段: vTESTstudio で実装したテストを CANoe で読み込み
 下段: CANoe で利用している通信データベースなどの情報を vTESTstudio から参照

1 つ目は、vTESTstudio で実装したテストを CANoe で読み込むことで、図 10 の中央上段の内容が該当します。vTESTstudio で実装されたテストは vTESTstudio 上で「ビルド」され、1 つのファイルにまとめられます。このファイルを CANoe から読み込むことで、テストを実行できるようになります。

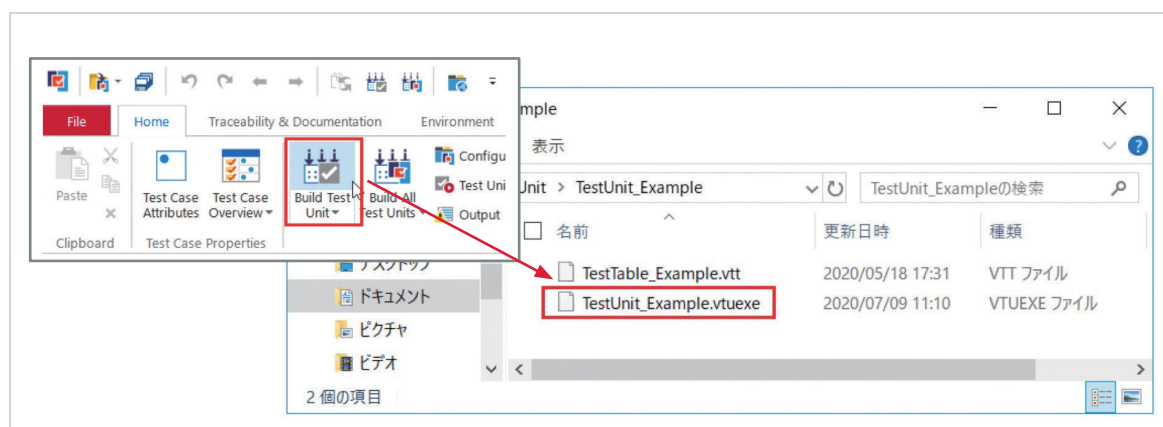


図 11: vTESTstudio で実装したテストは、ビルドすることで CANoe にて読み込み可能なファイルとなる。ビルド作業はボタン一つで完了

vTESTstudio 上のビルド作業も CANoe からの読み込み作業も、基本的にボタン 1 つでできるため、連携作業に工数は必要ありません。

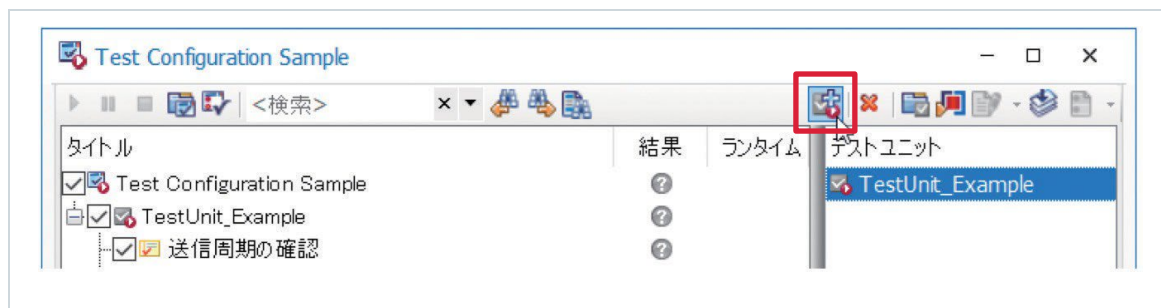


図 12: vTESTstudio でビルドしたテストシナリオのファイルを CANoe から読み込み。ボタン一つで読み込み作業が完了

2 つ目は、CANoe 上で利用するデータについて、vTESTstudio から参照することで、図 10 の中央下段部分の内容が該当します。vTESTstudio でテストを実装するには、テスト実行環境である CANoe と同じデータを参照する必要があります。ここでいうデータには、たとえば通信データベースの情報などが該当します。

この参照作業も、vTESTstudio 上のボタン一つで実行できるようになっています。

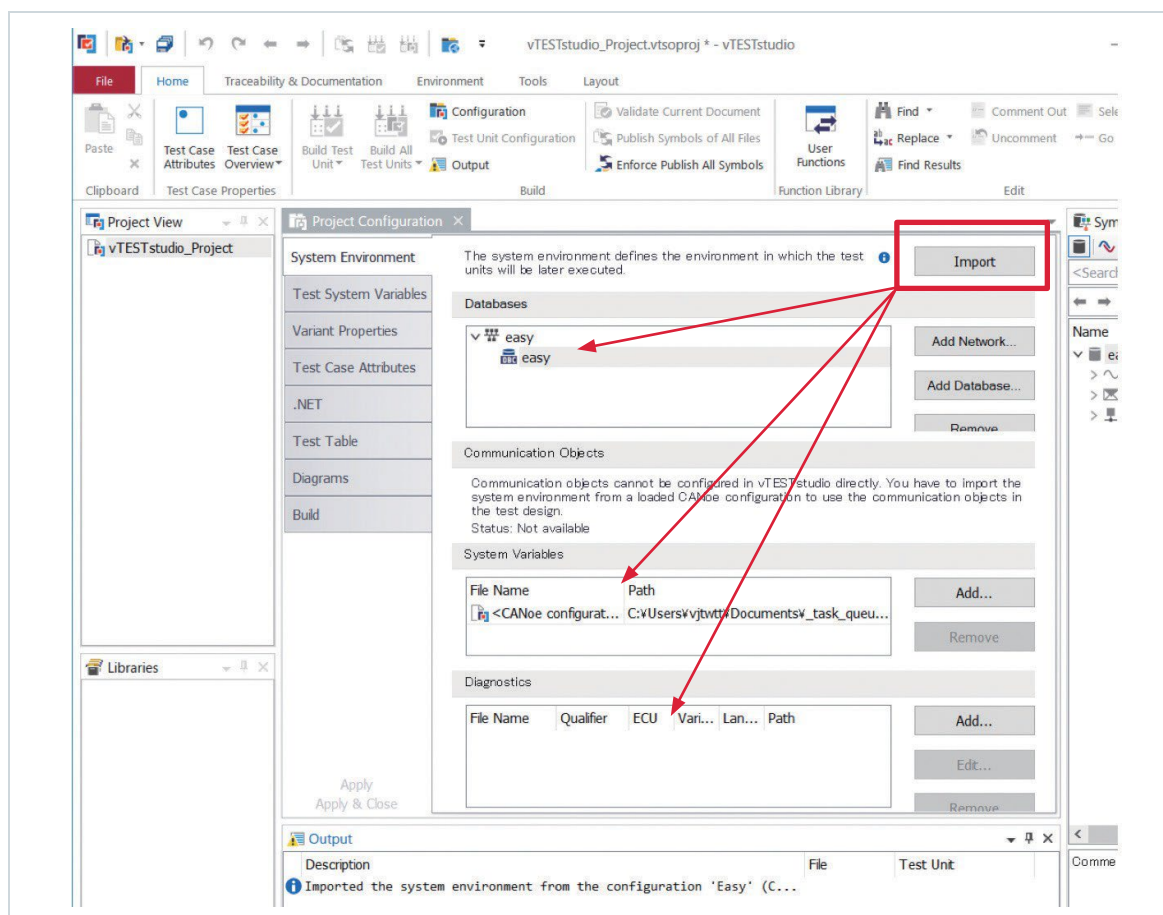


図 13: vTESTstudio 上のボタン一つで CANoe からデータをインポート可能

以上のように、vTESTstudio は CANoe とは独立したアプリケーションではありますが、CANoe 専用のツールであるため、ツールをまたいでいることを特に意識することなく、シームレスに連携することができます。そして、vTESTstudio の持つ、テスト実装を効率化するさまざまな機能を使用することで、CANoe 単体でテストを実装するよりも、実装にかかる工数を大幅に削減することができます。

vTESTstudio の具体的なテスト実装効率化機能については本稿 2 章にて紹介していきます。

1.3 VT システムの役割

ECU にはさまざまな入出力がありますが、以下の 4 つに分類できます。

1. CAN などの車載ネットワーク
2. センサーからの入力
3. アクチュエータへの出力
4. 電源

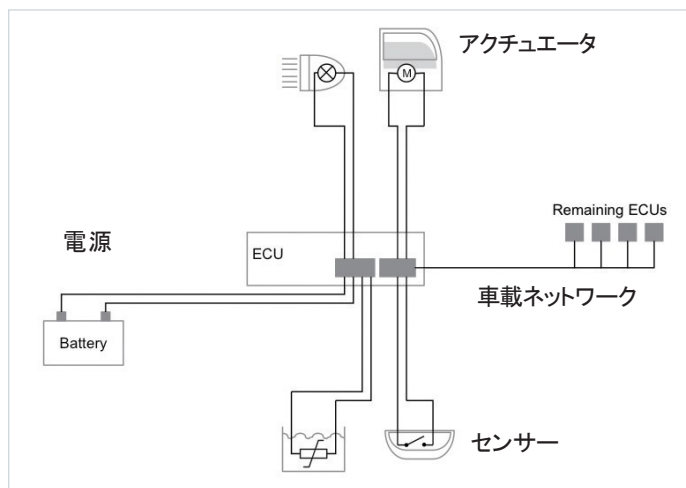


図 14: ECU の 4 種の入出力。ECU の統合テストにはこれらすべてを網羅する必要がある

本稿ではここまで、ECU の入出力のうち車載ネットワークについてのみ説明してきましたが、ECU のテストを実施するには他の入出力もすべて網羅する必要があります。VT システムは、車載ネットワーク以外の 3 つの入出力についても CANoe で制御するために作られた、車載 ECU テスト用のハードウェアとなります。VT システムは ECU の出力電圧値などを測定し CANoe に報告することや、CANoe から指示された値を電気信号に変換して ECU に刺激として入力することができます。

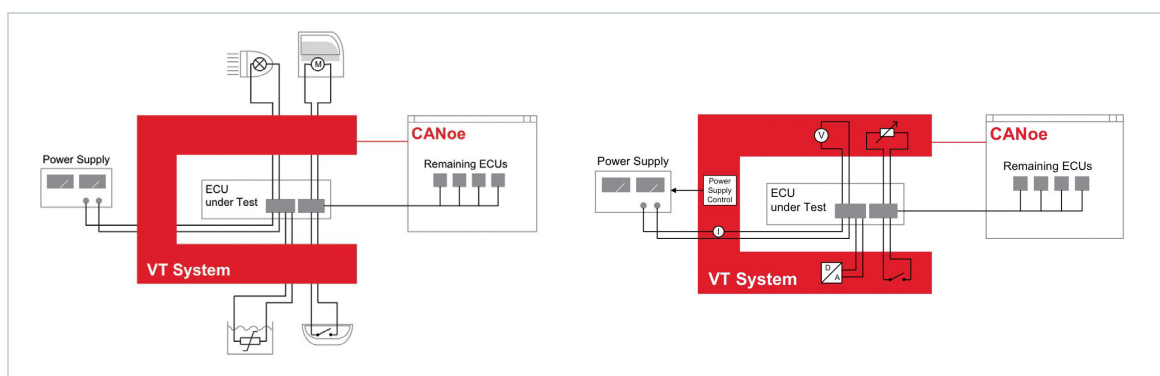


図 15: VT システムを使用すれば ECU の 4 種の入出力を網羅できる
 [左図] ECU と、実際のセンサーやアクチュエータなどとのやりとりを、VT システムがモニタリング
 [右図] センサーやアクチュエータの実機がない場合でも、VT システムが模擬可能

前述では ECU の入出力を 4 種に分類でき、VT システムを使用すればこの 4 種すべての入出力を網羅できることを説明しましたが、実際に必要となる入出力の種類(アナログ/デジタルやPWM など)やチャンネル数は、ECU 仕様に応じて大きく異なります。VT システムは、多様な ECU の仕様に柔軟に対応するために、モジュール形式のハードウェアとなっています。そのため、VT システムを構築する際は、ECU 仕様に合わせて必要なモジュールだけを組み合わせでシステム構築できるようになっています。また、後からモジュールを追加することも容易なため、別の ECU 開発への流用も低コストで行うことができます。



図 16: VT システムはモジュール形式のハードウェア。上記は 12 スロットタイプのケースを使用しているが、6 スロットタイプもある

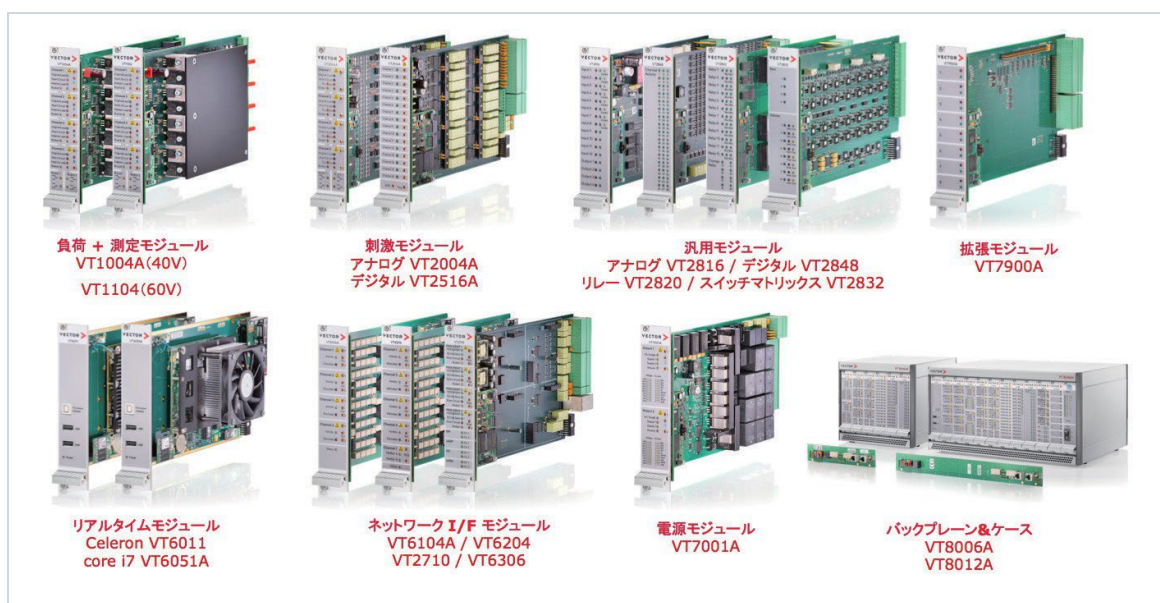


図 17: VT システムのモジュールラインナップ。用途に応じてさまざまなモジュールを提供しており、ECU 仕様に合わせて選択可能

VT システムについても前述の図 9 に加えると、以下の図 18 のようになります。これで CANoe、vTESTstudio、VT システムの 3 つの製品の関連性をすべて表すことができました。

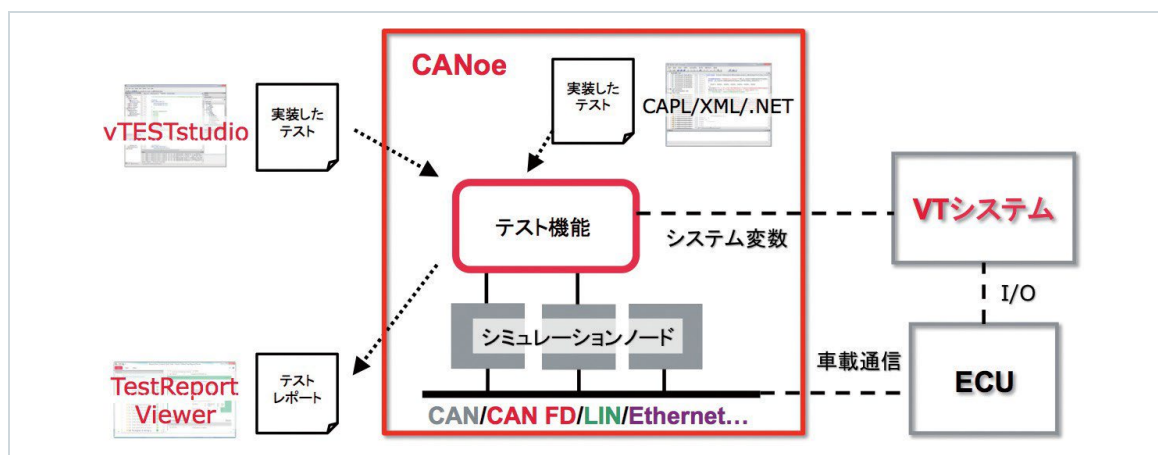


図 18: 3 つの製品の位置づけ、VT システムを追加し、3 つの製品すべてを記載

■ システム変数

上記の図 18 に記載しているとおり、CANoe と VT システムはシステム変数という CANoe 上の変数を介してデータ交換しています。システム変数は CANoe 上で自由にユーザー定義できる変数でもあり、VT システム用のデータだけでなくさまざまな用途に使用されます。

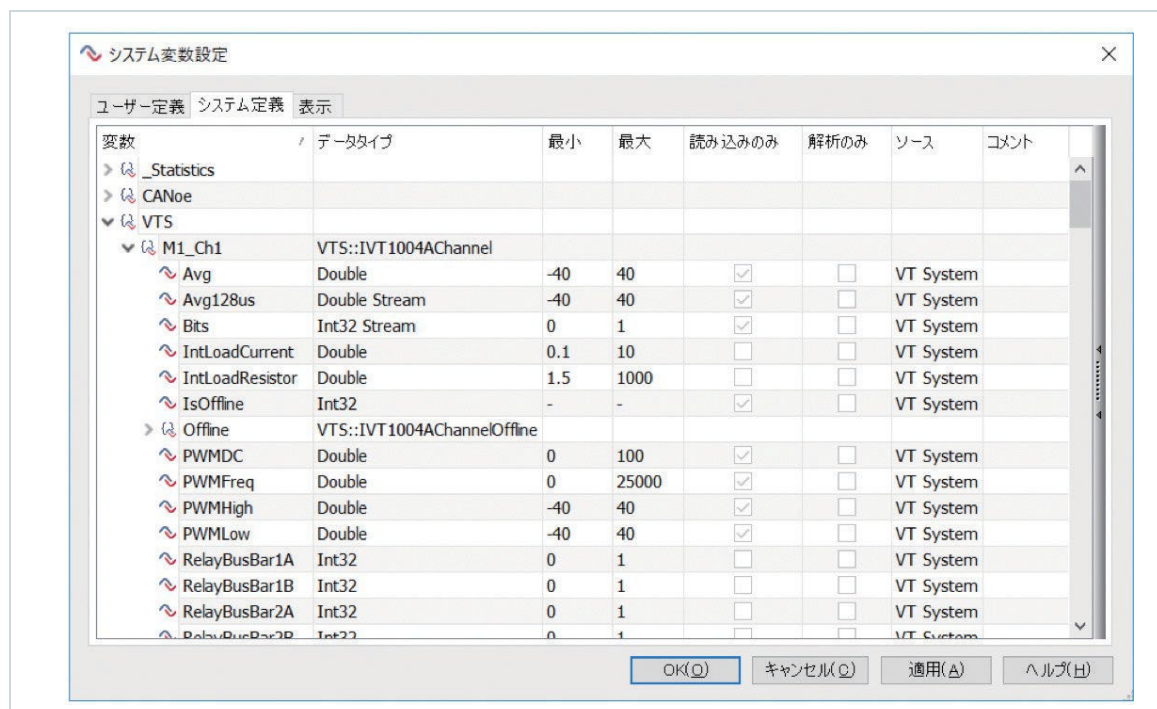


図 19: システム変数とは CANoe 上で利用できるグローバル変数。VT システム制御用のシステム変数は自動設定される

※VT システムの機能の詳細につきましては、V T システムに関する製品資料、またはアドバンスマニュアルを参照してください。

2. どんなことができるのか？

1 章ではベクターテストソリューションの概要と、vTESTstudio の役割について簡単に紹介しました。2 章では vTESTstudio のさまざまな機能について概要を説明し、vTESTstudio を使うとどんなことができるのか紹介していきます。

2.1 vTESTstudio で容易にできることの例

vTESTstudio は車載 ECU のテスト実装を効率化するためのツールです。そのため、ECU のテストを行う方が抱えるさまざまな問題に対応できるよう、多様な機能を提供しています。以下の表 B に vTESTstudio で容易に実現できることをいくつか例示します。

No.	実現したいことの例
1	CAN通信のチェックがしたい
2	診断通信のチェックがしたい
3	シグナル値の刺激に対する応答を確認したい
4	ECU内部値の動作をチェックしたい
5	アナログ／デジタルの入出力について、刺激に対する応答を確認したい
6	電源のON/OFFに対するECUの状態をチェックしたい
7	作成済みのテストを再利用したい
8	固有の仕様に基づく複雑なテストを作成したい
9	波形を定義してECUへの刺激として入力したい
10	テスト全体を一覧表示し、テストの抜け漏れチェックをしたい

表 B: vTESTstudio で容易に実現できることの例

もちろん、vTESTstudio で実現できることは上記の表 B に記載した内容以外にも数多くあります。表 B では、実際に vTESTstudio の導入を検討されたお客様からいただいたご質問やご要望の中から、多く寄せられるものを抜粋して記載しています。

本章ではこの表 B の内容をベースに vTESTstudio の機能を紹介していきます。

■ vTESTstudio が持つさまざまなエディター

vTESTstudio は、表 B のようなさまざまな要望に対応するため、多様な機能を持ちながらも使い勝手を損なわないよう、複数のエディターを内包しています。したがって、実現したい内容に応じて使用するエディターを適宜使い分けることができます。また、vTESTstudio ではこれらのエディターを単に使い分けるだけでなく、相互に連携して使用することも可能で、それぞれのエディターの良いところを組み合わせることができます。

vTESTstudio に含まれるエディターには、テスト設計エディターと、これをサポートするエディターの 2 種類があります。以下に表形式で記載します。

テスト設計エディター
Test Table Editor
Test Sequence Diagram Editor ※オプション「Graphical Test Design」が必要
State Diagram Editor ※オプション「Graphical Test Design」が必要
Coding Editor (CAPL、C#)

表 C: vTESTstudio が持つ、テストケースを作成するための 4 つのテスト設計エディター

テスト設計エディターをサポートするエディター
Parameter Editor
Waveform Editor
Classification Tree Editor

表 D: テスト設計を更に効率化するための 3 つの補助的なエディター

テスト設計エディターはその名のとおりに、テストのシナリオを作成するエディターとなります。vTESTstudio ではこのメインのエディターに加えて、実装を効率化するための 3 つの補助的なエディターを備えています。

本稿だけでこれらすべてのエディターの機能を紹介することはできませんので、これらのエディターを使用して表 B の内容をどのように実現していくのか、という形で 2.3 節以降にて紹介していきます。ここではまず、表 B の内容に vTESTstudio での実現方法を追記した以下の表 E を見てみましょう。

No.	実現したいことの例	vTESTstudioでの実現方法	参照節
1	CAN通信のチェックがしたい	Test Table Editorで定義済みコマンドの使用で容易に実現。「B ackground Checks」に属するコマンド群を使用	2.3節
2	診断通信のチェックがしたい	Test Table Editorで定義済みコマンドの使用で容易に実現。「Diagnostics」に属するコマンド群を使用	2.4節
3	シグナル値の刺激に対する応答を確認したい	Test Table Editorで定義済みコマンドの使用で容易に実現。「State Change」コマンド1つで実装可能	2.5節
4	ECU内部値の動作をチェックしたい	CANoe.AMD/XCPを使用して内部値をシステム変数で制御。vTES Tstudio上ではNo.3と全く同じ	
5	アナログ／デジタルの入出力について、刺激に対する応答を確認したい	VTシステムを使用しアナログ／デジタルの入出力データをシステム変数で制御。vTESTstudio上ではNo.3と全く同じ	
6	電源のON/OFFに対するECUの状態をチェックしたい	No.5と全く同じ	
7	作成済みのテストを再利用したい	Test Table Editorの「Function Definition」や「Test Case Definition」、およびParameter Editorのパラメーター定義で汎用的・網羅的なテストを生成	2.6節
8	固有の仕様に基づく複雑なテストを作成したい	Test Table EditorとCoding Editorを連携し、複雑な部分のみCAPL などのプログラミングを使用する	2.7節
9	波形を定義してECUへの刺激として入力したい	Waveform Editorに刺激用の波形を定義し、Test Table Editorでこの波形を呼び出して使用	2.8節
10	テスト全体を一覧表示し、テストの抜け漏れチェックをしたい	Test Sequence Diagram Editorでテストの全体像を描画	2.9節

表 E: vTESTstudio で容易に実現できることの例。vTESTstudio での実現方法を追加

表 E で追加された vTESTstudio での実現方法の列を見ると、Test Table Editor を使用して実現する項目が大半となっています。Test Table Editor は vTESTstudio の中でも最も標準的で、最も使用されるエディターです。そのため、表 E の各項目の説明に移る前に、次の 2.2 節にて Test Table Editor の概要を簡単に説明します。

2.2 Test Table Editor の概要

vTESTstudio の編集画面は以下の図のようになります。

画面上部にリボン形式のメニューが並び、その下に各種編集用のビューが配置されます。編集用のビューは好きな場所に配置することができますが、デフォルトでは画面左側にプロジェクトビューやライブラリー、画面中央に各エディターの編集画面、画面右側にテストコマンドやシンボル、画面下段に出力情報が配置されています。

図 20 では画面中央に Test Table Editor 編集画面を表示しています。他のエディターを使用する場合にも、同じ領域に編集画面が表示されます。

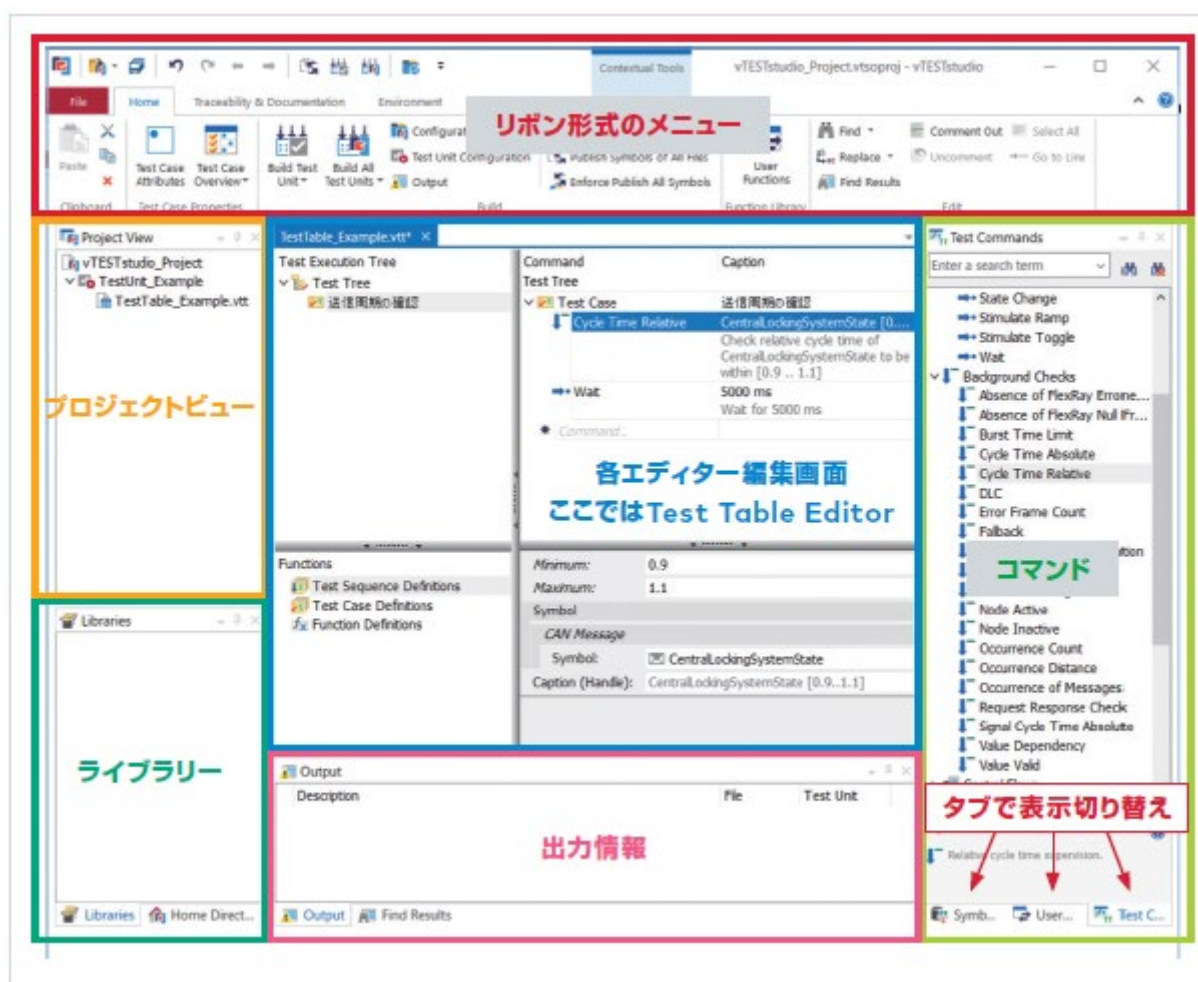


図 20: vTESTstudio 編集画面 概観

■ Test Table Editor によるテストケースの実装

Test Table Editor はコマンドを並べるだけで直感的にテストを実装できる優れたエディターです。プログラミング経験のない方でもコマンドやシンボル(CANoe、vTESTstudio ではメッセージやシグナル、システム変数などの変数を総称して「シンボル」と呼称します)のドラッグ&ドロップや、値の入力を行うだけで容易にテストを実装することができるため、少ない工数でテストを作成できます。

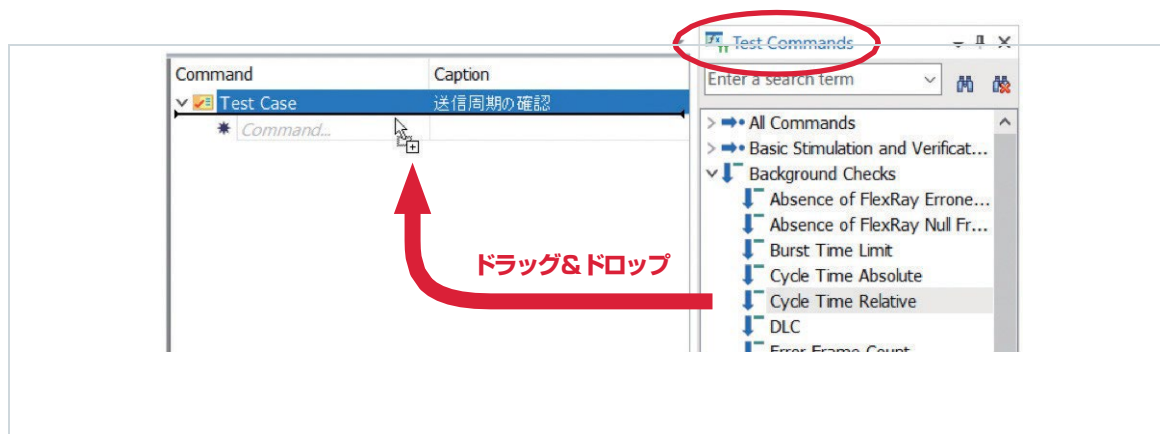


図 21-1: Test Table Editor 上でテストを実装: コマンドをドラッグ&ドロップで追加

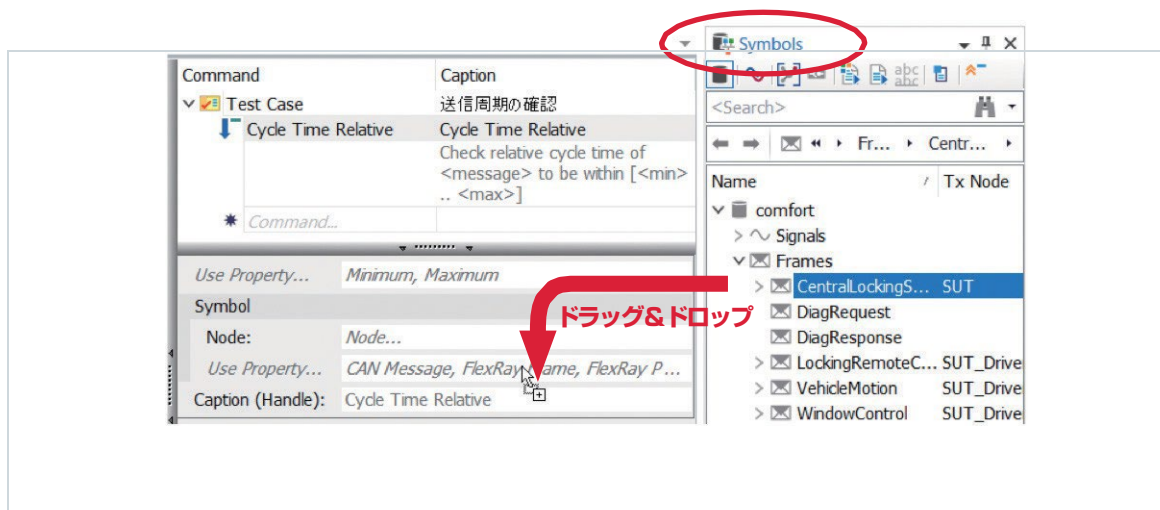


図 21-2: Test Table Editor 上でテストを実装: 対象のシンボルをドラッグ&ドロップで追加

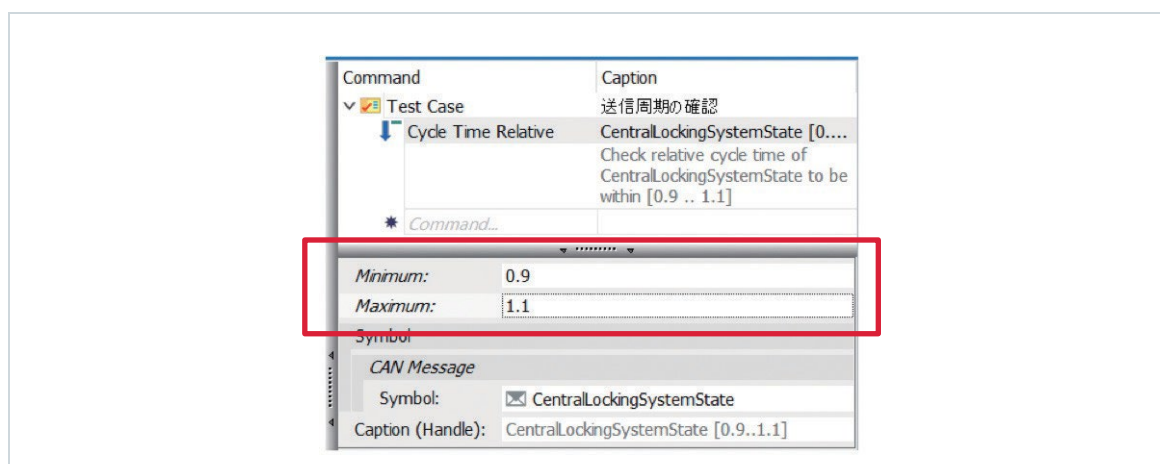


図 21-3: Test Table Editor 上でテストを実装: コマンドの属性値などを設定

vTESTstudio 上ではテストのひとかたまりを「テストケース」と呼びます。そしてそのテストケース内で行われる処理 1 つ 1 つを「テストステップ」と呼びます。このテストステップは、前頁の 3 つの図で見たようにテストケース内に定義済みのコマンドを配置し、対象のシンボルや必要な設定値を追加するだけで作成できます。そして、このテストステップを図 21-4 のようにテストケース内に並べていくだけで、テストの実装が完了します。つまり、前頁の 3 つの図で示したように、コマンドの配置、シンボルや設定値の追加といった作業を繰り返すだけで、テストケースが作成できるようになっています。

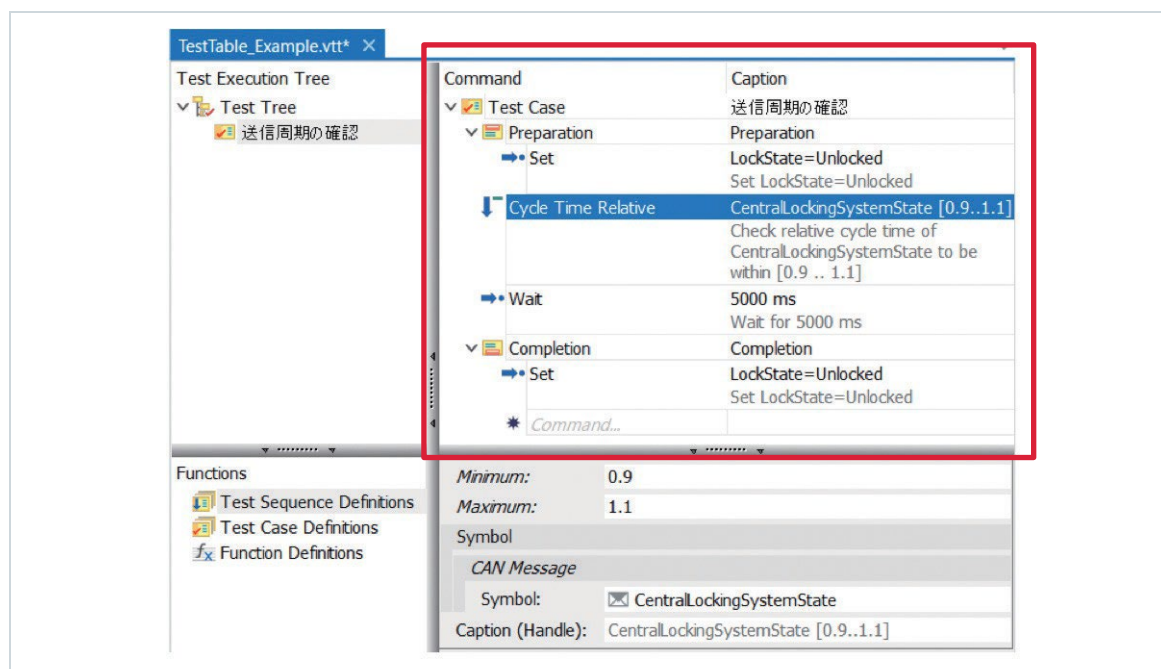


図 21-4: Test Table Editor 上でテストを実装: コマンドを並べていくだけでテスト実装が完了

Test Table Editor で実装したテストケースを CANoe で実行すると、それぞれのコマンドが上から順番に実行されていき、各コマンドの実行結果や判定結果がレポートに記録されます。レポートに出力される内容はコマンドごとに最適な内容があらかじめ定義されているため、レポートへ出力する内容を別途実装する必要はありません。

■ 多数のテストケースの管理

前述のとおり、テストケースはテストのひとかたまりです。そのため、実際に ECU のテストを実装する際には数多くのテストケースを作成することになります。Test Table Editor で作成したテストケースは、図 21-4 の画面左上にある「Test Tree」の下に配置しますが、テストケースの数が多くなると、どのテストがどこにあるのか分かりづらくなります。これに対し、Test Table Editor はテストケースのグループ化などの構造化機能を持っており、多数のテストケースの管理にも容易に対応できるようになっています。ここでは詳細は割愛しますが、ご興味のある方は、vTESTstudio のトレーニングや vTESTstudio に関するアドバンスマニュアルを参照してください。

※なお、「Test Tree」はテストケースの集合体における最上位の階層を示す、フォルダのようなものです。「Test Tree」自体にはテストは実装せず、テストケースなどを配下に置くための便宜的な存在ですので、名称を覚えていただく必要はありません。

それでは、2.1 節の表 E について、それぞれの実現方法を次節以降で見いていきましょう。

2.3 CAN 通信のチェック

表 E の No.1 に記載しています CAN 通信のチェックは、ECU が CAN 通信を行う場合に汎用的に行われるテストです。たとえば、ECU の送信するメッセージの送信周期のブレ幅や、CAN のエラーフレーム発生状況など、CAN 通信が正しく実施できているか、確認します。このような何度も繰り返し行われる、人間が行うには退屈なテストは、自動化と相性が良いものです。そのため vTESTstudio では専用コマンドを用意しており、容易に実装できるようになっています。

通信が正しく行われているか？というテストは以下のように ECU 同士の通信や、ECU とテスター(=CANoe)との通信をモニタリングする形で行われます。

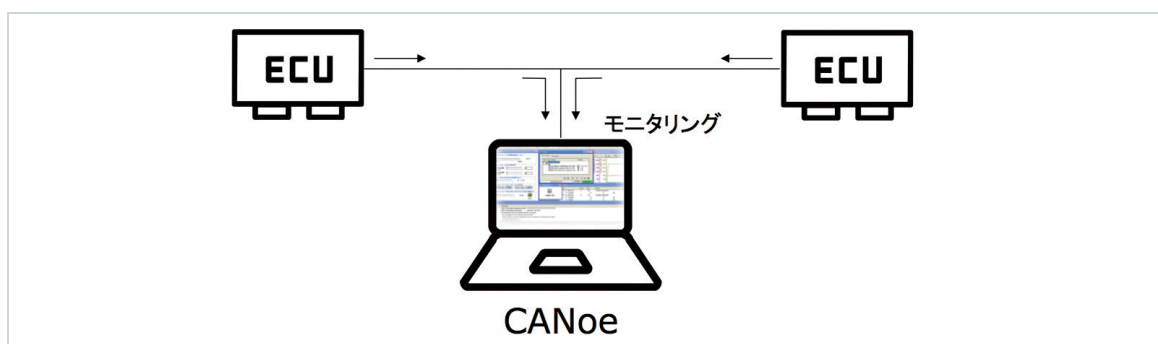


図 22: ECU の CAN 通信が正しく行われるか CANoe が監視

このような監視を行うコマンド群として、vTESTstudio では「Background Checks」という定義済みコマンドのグループがあります。この中にあるコマンドのうち、よく使われるものを以下に例示します。

コマンド名	内容、用途
Cycle Time Relative	ECUの送信するメッセージの送信周期を監視する。このコマンドの場合、データベースに定められた周期に対する比率(例:90%~110%に収まっているか?など)でブレ幅を指定する。周期の絶対値で指定するコマンドとしてCycle Time Absoluteもある
DLC	ECUの送信するメッセージのDLCがデータベースどおりか監視する
Error Frame Count	バス上にエラーフレームが発生していないか監視する
Message Known	バス上に未定義のメッセージが送信されていないか監視する
Value Valid	ECUから送信されるシグナル値が正常な範囲内にあるかを監視する。似たコマンドにNo Value Changeがある
Node Active	対象のノード(ECU)がメッセージ送信しているか監視。電源起動後一定時間でECUが通信開始するか確認するときなどに使用。対になるコマンドにNode Inactiveがある

表 F:「Background Checks」に含まれる定義済みコマンドの例

CAN 通信のチェックはこれら「Background Checks」のコマンドを使用するだけで容易に実現できます。Test Table Editor 上で実装が完了したテストケースの一例を以下に示します。

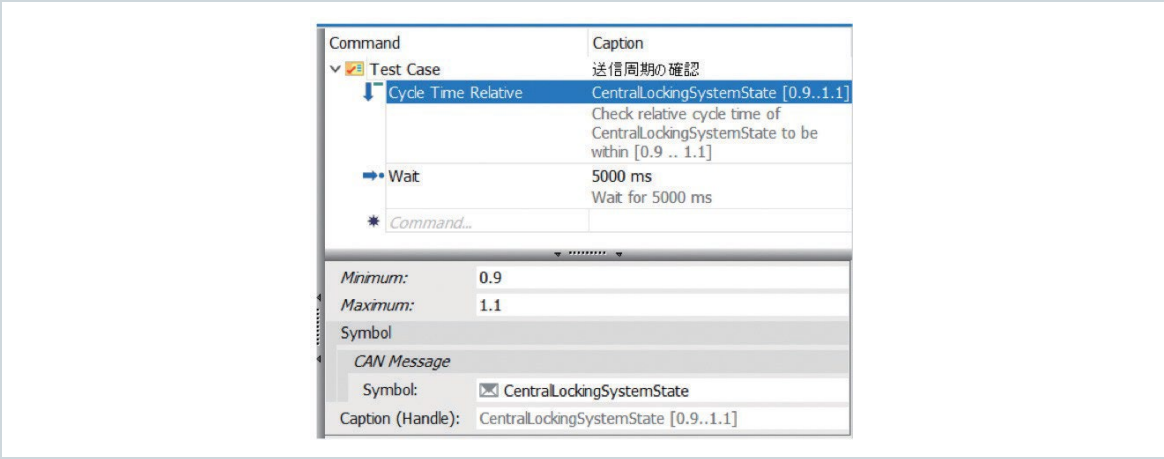


図 23: CAN メッセージの送信周期を確認するテスト実装例。専用コマンドを使用するだけで実装完了

上記のテストケースでは、「CentralLockingSystemState」という CAN メッセージに対し、送信周期のブレ幅が 90%～110%以内に収まっているか監視しています。監視時間は「Wait」コマンドにて 5000ms と指定しています。

このテストを CANoe で実行し、自動生成されるレポートを確認すると以下のように送信周期の統計も表示されます。vTESTstudio ではこのように、CAN 通信のチェックを非常に容易に実現できます。

Statistics

Summary	
Statistic	
Runtime of the statistic	5000 ms
Number of samples	50
Message ID	257 (0x101)
Minimal measured cycle time	99.456 ms
Maximal measured cycle time	100.799 ms
Average cycle time	100.017653 ms
Failure ratio (in %)	0

Distribution of measured times																				
<= 80 ms	80 ms - 82 ms	82 ms - 84 ms	84 ms - 86 ms	86 ms - 88 ms	88 ms - 90 ms	90 ms - 92 ms	92 ms - 94 ms	94 ms - 96 ms	96 ms - 98 ms	98 ms - 100 ms	100 ms - 102 ms	102 ms - 104 ms	104 ms - 106 ms	106 ms - 108 ms	108 ms - 110 ms	110 ms - 112 ms	112 ms - 114 ms	114 ms - 116 ms	116 ms - 118 ms	> 120 ms
0	0	0	0	0	0	0	0	0	0	31	18	0	0	0	0	0	0	0	0	0

図 24: Cycle Time Relative コマンドを使用した際にレポートに出力される、送信周期の統計情報

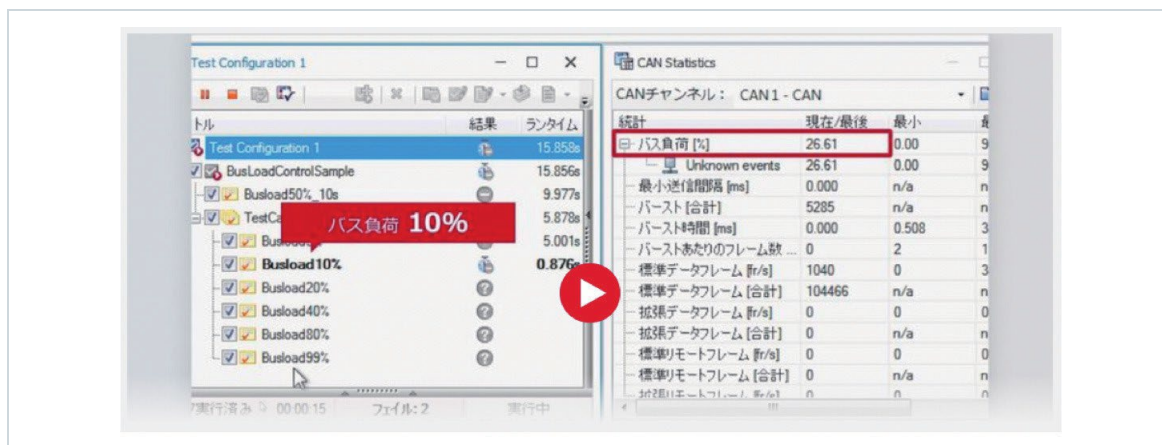
■ Cycle Time Relative コマンドの後ろに Wait コマンドが配置される理由

ここで紹介した「Background Checks」のコマンドは、名前のとおりバックグラウンドで動作するコマンドとなります。vTESTstudio ではこのことを端的に示すため、他のコマンドと異なるアイコンが使われています。「Background Checks」は  というアイコンで、「監視をバックグラウンドで開始する」という意味になります。これに対し、その直後で使用されている Wait コマンドなどの通常のコマンドは、 というアイコンで、「その場で処理をする」という意味を示しています。「Background Checks」のコマンドは上述のとおりに、監視を開始することを示していますので、コマンド 1 つだけでテスト実装完了とはできず、その後ろに Wait コマンドなど実際に待機時間を設定可能な処理を配置し、監視時間を設ける必要があります。

vTESTstudio で CAN 通信に関するテストを実際に作成する際には、弊社ベクター・ジャパン Member Area で公開しています、「CANoe テスト機能、vTESTstudio 自習用サンプル」が参考になります。このページから vTESTstudio の基本的な操作方法を学習するためのトレーニングテキストと、付属のサンプルがダウンロードできます。トレーニングテキストに従ってサンプルを操作することで、CAN 通信に関するテストの詳しい実装方法を確認することができます。是非ご覧ください。

 04.CANoeテスト機能、vTESTstudio自習用サンプル	--	06.08.2020 06:42	04.09.2020 02:58	CANoeテスト機能、vTESTstudioの基本的な使い方を学習するためのトレーニングテキスト、サンプルをダウンロードいただけます。
--	----	---------------------	---------------------	---

また、本節で説明した CAN 通信のチェックを、CAN ネットワークに高い通信負荷をかけた状態で行う、CAN のバス負荷に関するテストも広く行われています。vTESTstudio には CAN のバス負荷を直接コントロールするコマンドはありませんが、CAPL の機能で容易に実現できるため、弊社ベクター・ジャパン Member Area の「サンプルコンフィギュレーション」にてバス負荷調整のテストのサンプルと適用方法を公開しています。ご興味ある方は参照してください。



2.4 診断通信のチェック

表 E の No.2 に記載しています診断通信のチェックも、ECU が診断通信を行う場合に汎用的に行われるテストです。診断通信の場合、診断リクエストを ECU に送り、正しい応答が返ってくるかを確認するテストが大半で、場合によっては CANoe のオプション製品である CANoe.Diva により診断データベースから診断テストを自動生成できることがあります。CANoe.Diva で生成するテストは網羅的で有用ですが、機能テストと併せて診断テストを行う場合や、ちょっとした確認を行いたい場合などには vTESTstudio でテストを作成することも有効です。

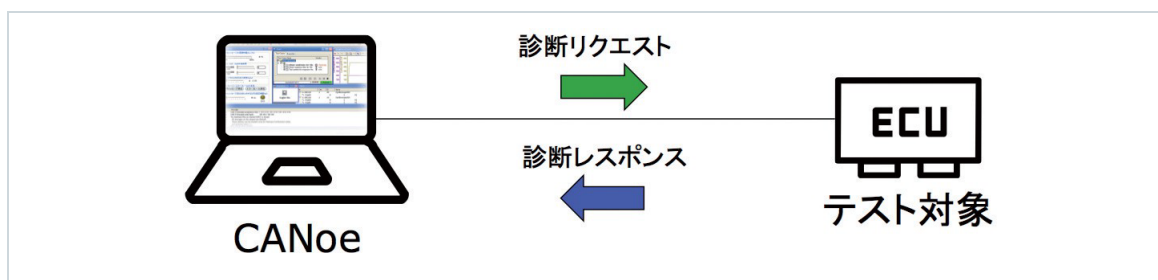


図 25: ECU の診断通信が正しく行われるか、CANoe と診断通信を行い確認

診断通信のチェックも CAN 通信のチェックと同様、何度も繰り返し行われる、人間が行うには退屈なテストであり、自動化と相性が良いものです。そのため vTESTstudio では専用コマンドを用意しており、容易に実装できるようになっています。

診断通信のテストを行うコマンド群として、vTESTstudio では「Diagnostics」という定義済みコマンドのグループがあります。この中にあるコマンドのうち、よく使われるものを以下に例示します。

コマンド名	内容、用途
Diagnostic Service	診断リクエストを送信し、レスポンスが期待したとおりか確認します。このコマンド1つで、多くの診断テストが実現できます
Diagnostic Unlock ECU	診断セッションの遷移を行います。事前にCANoeにSeed & Key用DLLを設定しておく必要があります
Fault Memory DTCs	DTCの状態をチェックします。DTCに関しては、このコマンドの他にもFault Memory ClearやFault Memory DTC Supportなどいくつかの有用なコマンドがあり、これらを組み合わせるとDTC関連のテストが容易に実現できます

表 G: 「Diagnostics」に含まれる定義済みコマンドの例

これらのコマンドの使用には、CANoe 上で診断サービスが定義されていることが前提となります。そのため、CANoe 上で診断データベースを割り付けるなどの設定が必要ですが、これら設定も容易に行うことができます。診断サービスが定義されていれば、vTESTstudio 上では診断通信のチェックはこれらのコマンドを使用するだけで容易に実現できます。

vTESTstudio で診断通信に関するテストを実際に作成する際には、弊社ベクター・ジャパン Member Area で公開しています、「CANoe テスト機能、vTESTstudio 自習用サンプル」が参考になります。このページから vTESTstudio の基本的な操作方法を学習するためのトレーニングテキストと、付属のサンプルがダウンロードできます。トレーニングテキストに従ってサンプルを操作することで、診断通信に関するテストの詳しい実装方法を確認することができます。是非ご覧ください。

 04.CANoeテスト機能、 vTESTstudio自習用サンプル	--	06.08.2020 06:42	04.09.2020 02:58	CANoeテスト機能、vTESTstudioの基本的な使い方を学習するためのトレーニング テキスト、サンプルをダウンロードいただけます。
--	----	---------------------	---------------------	---

2.5 刺激に対する応答を確認

表 E に記載の No.3 から No.6 までは、すべて「刺激に対する応答」というパターンとみなすことができます。そのため、本節でまとめて説明します。このパターンのテストは普遍的で、ECU に対するブラックボックステストとして非常に頻繁に行われます。

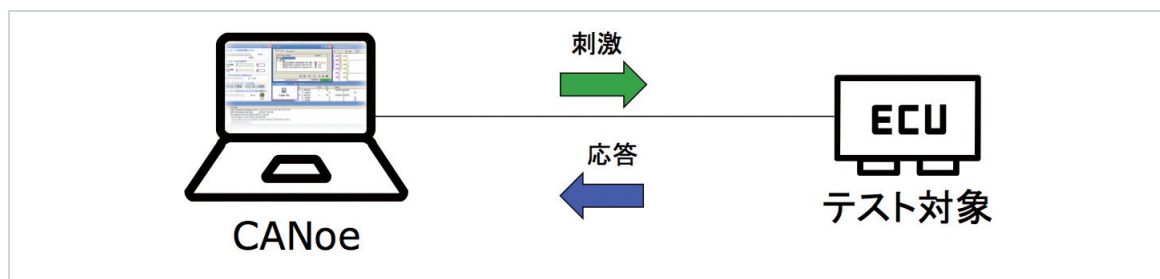


図 26: ECU に対する典型的なブラックボックステスト。刺激に対する応答を確認

以下に No.3 から No.6 までのテストについて、それぞれのテストの具体的な内容を説明します。

No.	実現したいことの例	具体的な内容
3	シグナル値の刺激に対する応答を確認したい	テスト対象ECUが受信するCANシグナルなどに値を入力し、この値に対するECUの応答、たとえば ECU が送信する別のCANシグナルの値が期待どおりかを確認
4	ECU内部値の動作をチェックしたい	CANシグナルなどでECUに刺激を与え、XCP通信経由でECU内部値を確認し、内部動作が正しく行われているか確認する。またこれとは逆に、内部値に刺激を与えて、ECUから送信するCANシグナルなどを確認するテストもよく実施される
5	アナログ／デジタルの入出力について、刺激に対する応答を確認したい	VTシステムなどのHWを用いてセンサーの模擬をし、ECUのセンサー入力に刺激を与える。この刺激に対してECUが正しくアクチュエータへの出力を行っているか確認
6	電源のON/OFFに対する ECU の状態をチェックしたい	電源のON/OFFを制御し、ECUの消費電流の変化や、電源起動後のECUの起動時間、通信状況などを確認。また、電源電圧の変動に対するECUの挙動を確認するなど

表 H: 表 E に記載の項目のうち、本節で説明する内容のみ抜粋し、再掲

ECU 内部値の取得については、CANoe.AMD/XCP を使用することで容易に実現できます。CANoe.AMD/XCP は CANoe のオプション製品で、内部値を定義した A2L ファイルを読み込み、対応するシステム変数を CANoe 上に自動生成する機能を持っています。また、このシステム変数の値に基づき、CANoe.AMD/XCP が XCP 通信を自動で実施します。そのため内部値の取り扱いは、対応するシステム変数への値の代入や、変数値のチェックを行うだけでよく、容易に制御できます。

アナログ／デジタルの入出力の値、電源電圧や消費電流などは、VT システムを用いて測定・制御を行うことができます。これらの I/O の値は CANoe 上ではシステム変数として扱われ、システム変数への値の代入や、変数値のチェックを行うことで容易に制御できます。なお、VT システム用のシステム変数は、CANoe 上で VT システムの設定を行うと自動的に生成されます。

システム変数については本稿の 1.3 節で簡単に説明しているとおり、車載ネットワークで用いられるシグナル以外の各種データを取り扱うための変数で、上記の ECU 内部値や I/O の値の他にもさまざまな用途に使用されます。

CANoe 上でシステム変数として扱うことができるようになったデータは、vTESTstudio 上で CAN シグナルなどと同様に扱うことができます。したがって、内部値のテストや I/O のテストなどについては、ハードウェアや CANoe 上の設定がきちんと実施されているかどうか、つまりテスト実行環境の整備ができているかがポイントとなります。この環境整備さえできていれば、テスト実装ツールである vTESTstudio は CANoe 上で扱えるデータに対してテスト実装できるようになっています。

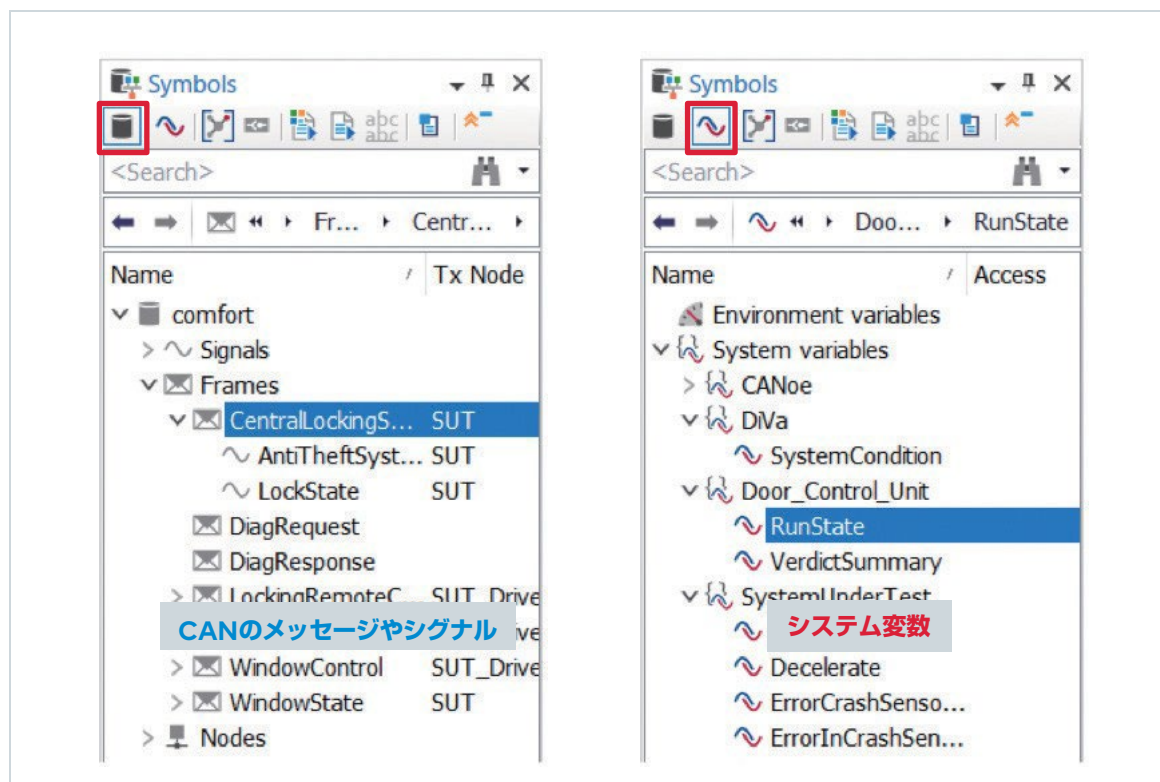


図 27: vTESTstudio 上では CAN シグナルもシステム変数も同様に取り扱い、テスト実装が可能

なお、CANoe にはさまざまな外部機器やサードパーティー製ツールと接続できるインターフェイスを多数持っており、内部値や I/O 以外のテストにも広く対応できるようになっています。

※CANoe.AMD/XCP や VT システム、外部機器連携などの機能の詳細につきましては、それぞれのアドバンスドマニュアル等を参照してください。

■ vTESTstudio で「刺激に対する応答」のテストを実装

「刺激に対する応答」というパターンのテストを vTESTstudio で実装するには、「State Change」コマンド 1 つで実現できます。「State Change」コマンドは「Set」「Wait」

「Check」という、より単純な 3 つのコマンドで構成されたもので、あまりにも頻繁に使用されるパターンであるため、1 つのコマンドとしても定義されています。

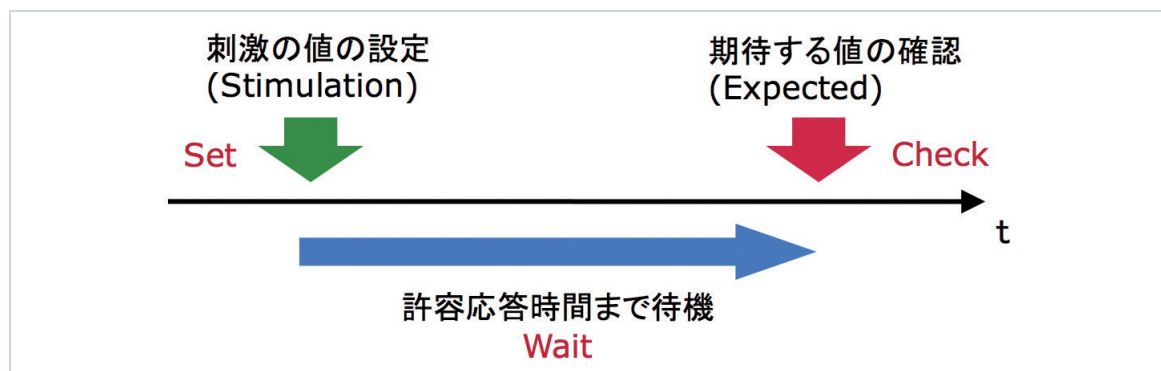


図 28: State Change コマンドは「Set」「Wait」「Check」の 3 つのコマンドを統合したもの

「State Change」コマンドを使用して Test Table Editor にてテスト実装した例を以下に示します。下記の例はシンプルな形ではありますが、テストケース内にコマンドをたった一つ配置し、値を設定するだけで、ECU に対する典型的なテストの実装ができてしまうことを示しています。

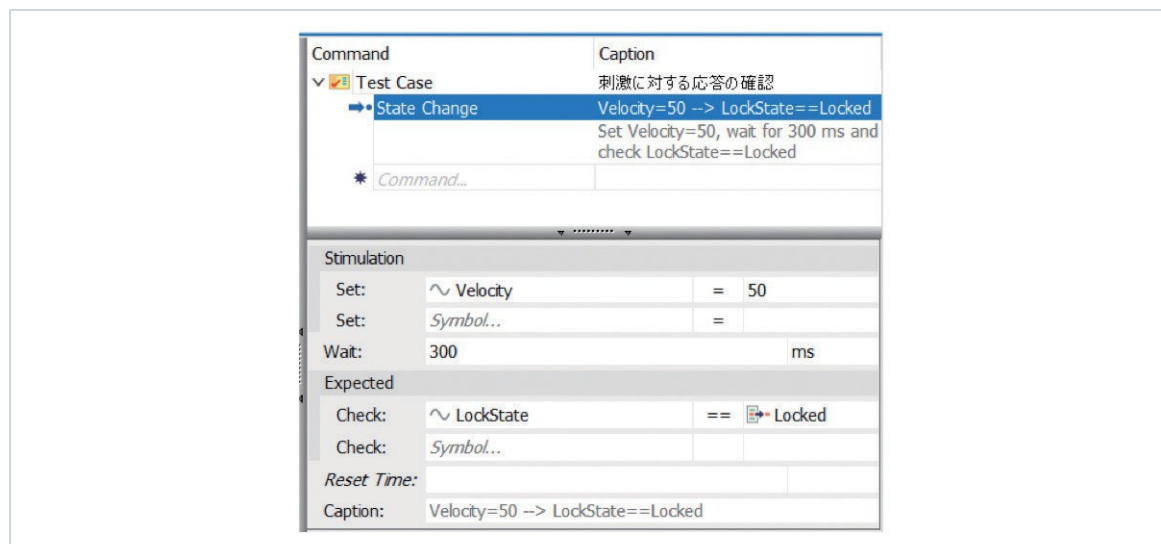


図 29: State Change コマンドを用いた「刺激に対する応答」を確認するシンプルなテスト実装例。1 つのコマンドで実現可能

なお、図 29 の「Expected」-「Check:」の部分に「Locked」という値の表記があります。これは値テーブルと呼ばれ、実態は数値です。値テーブルとは、値の意味が分かりやすいようラベルが貼られたデータで、CANoe や vTESTstudio では値の代わりにこのラベルを表示させることができます。

上記のテストを実際に作成する際には、弊社ベクター・ジャパン Member Area で公開しています、「CANoe テスト機能、vTESTstudio 自習用サンプル」が参考になります。このページから vTESTstudio の基本的な操作方法を学習するためのトレーニングテキストと、付属のサンプルがダウンロードできます。トレーニングテキストに従ってサンプルを操作することで、「刺激に対する応答」を確認するテストの詳しい実装方法を確認することができます。是非ご覧ください。

 04.CANoeテスト機能、vTESTstudio自習用サンプル	--	06.08.2020 06:42	04.09.2020 02:58	CANoeテスト機能、vTESTstudioの基本的な使い方を学習するためのトレーニングテキスト、サンプルをダウンロードいただけます。
--	----	---------------------	---------------------	---

2.6 作成済みのテストの再利用

表 E の No.7 に記載のテストの再利用は、自動テストにおいて非常に重要です。なぜなら、テスト実装の工数を削減する最も有効な手段が、テストの再利用だからです。vTESTstudio はこの再利用に関するさまざまな機能を備えています。ここではその中からいくつか良く使用される機能について紹介します。

2.6.1 テストケースを再利用する

作成済みテストの再利用について、まずはテストケースを再利用する機能を紹介します。テストケースの再利用について説明する上で、ここでは前節の「刺激に対する応答」を確認するテストを題材にします。このパターンのテストは通常、刺激と応答の値のペアは 1 組だけでなく複数の組に対して実施されます。また、場合によっては数千組から数万組といった膨大な数の組み合わせについても実施されるケースがあります。

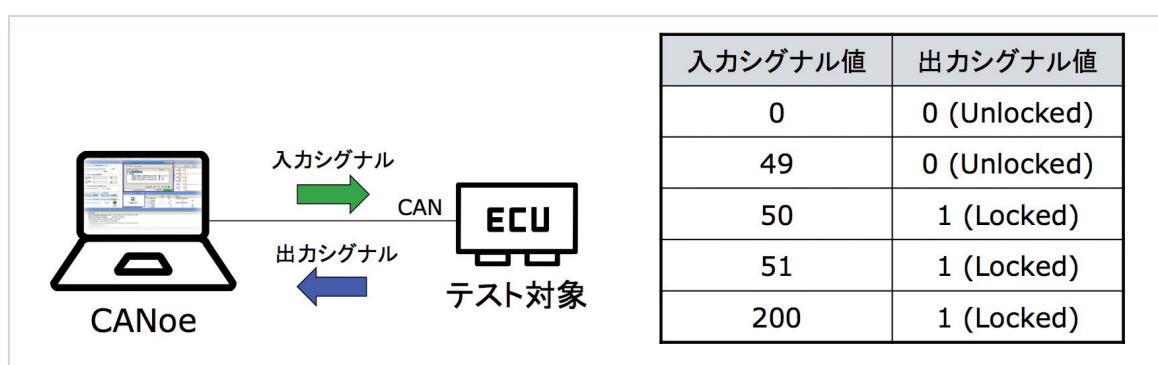


図 30:「刺激に対する応答」の確認は、実際には多数の値の組み合わせをチェックすることが多い
※なお「Unlocked/Locked」は値テーブルと呼ばれ、実態は数値です。詳細は前ページを参照してください。

前節で述べたように「State Change」コマンドを使えば、テストケースは簡単に実装できます。しかし、すべての値のペアに対して毎回テストケースを作成するのはやはり工数がかかります。vTESTstudio ではこういった場合に対応できるよう、「Test Case Definition」と、テストケースリストという機能を持っています。

■ Test Case Definition

vTESTstudio では Test Table Editor 上で「Test Case Definition」というテストケースのひな型を定義することができます。作成方法は通常のテストケースと同様ですが、通常のプログラミング言語でいうところの関数のように、引数を受け取り、テストケースを汎用化することができます。

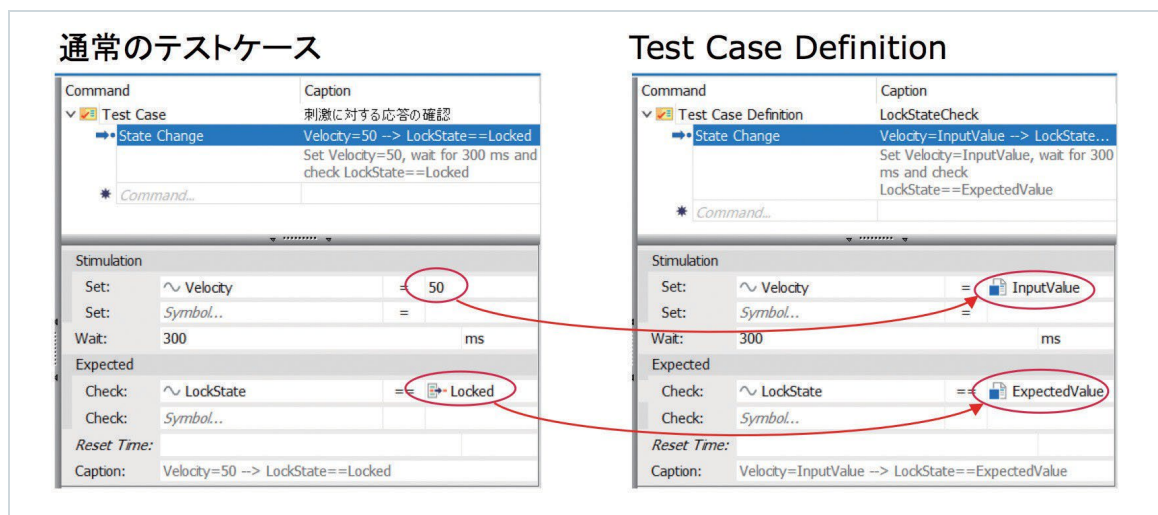


図 31: 通常のテストケースでは直接値を指定している一方、Test Case Definition では引数となり、外部から柔軟に値を設定可能

この「Test Case Definition」を使用するときには、プログラミングにおける関数のように、引数に値を指定して呼び出します。「Test Case Definition」を使用してひな型を作成しておくことで、同じパターンのテストを毎回実装することなく、何度でも同じテストケースを利用できるようになります。

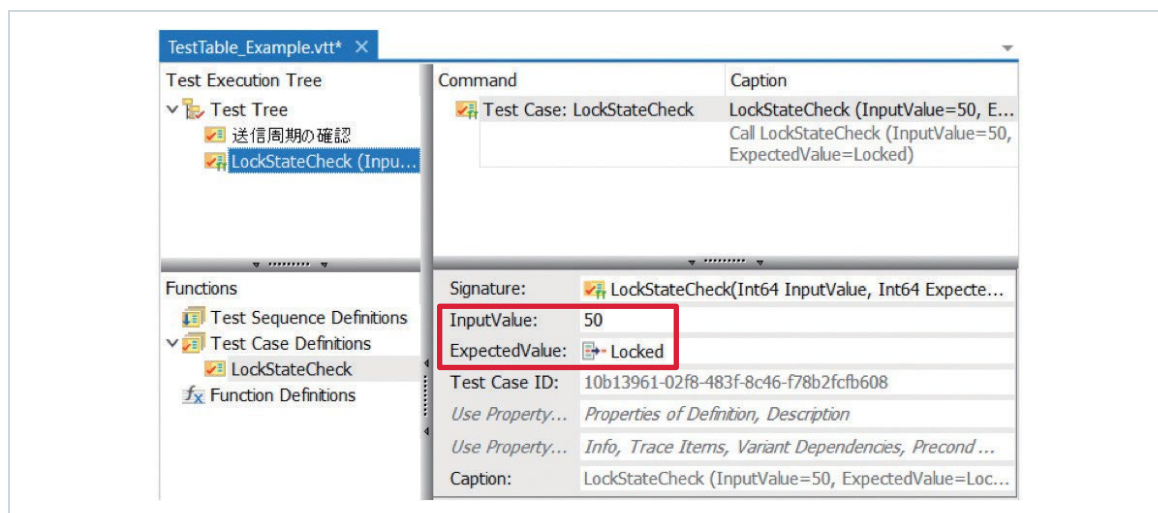


図 32: Test Case Definition を呼び出し。引数に値指定するだけで、テストケースの作成が完了

■ テストケースの抽出

実際のテスト実装プロジェクトでは、テストの実装が進み、ある程度の数のテストケースができあがった段階で、ようやく必要なひな型が判明する、ということが頻繁に起こります。つまり、テスト実装プロジェクトの開始時点ではどのようなひな型が必要になるのか不明なことが多く、個別のテストケースの実装開始前に Test Case Definition を作成することは難しい、と言えます。

しかし、一旦通常のテストケースを作った後で、再度ひな型である Test Case Definition を作成し直すのは若干工数がかかります。このような場合のために、vTESTstudio ではテストケースから Test Case Definition を抽出する機能を持っています。

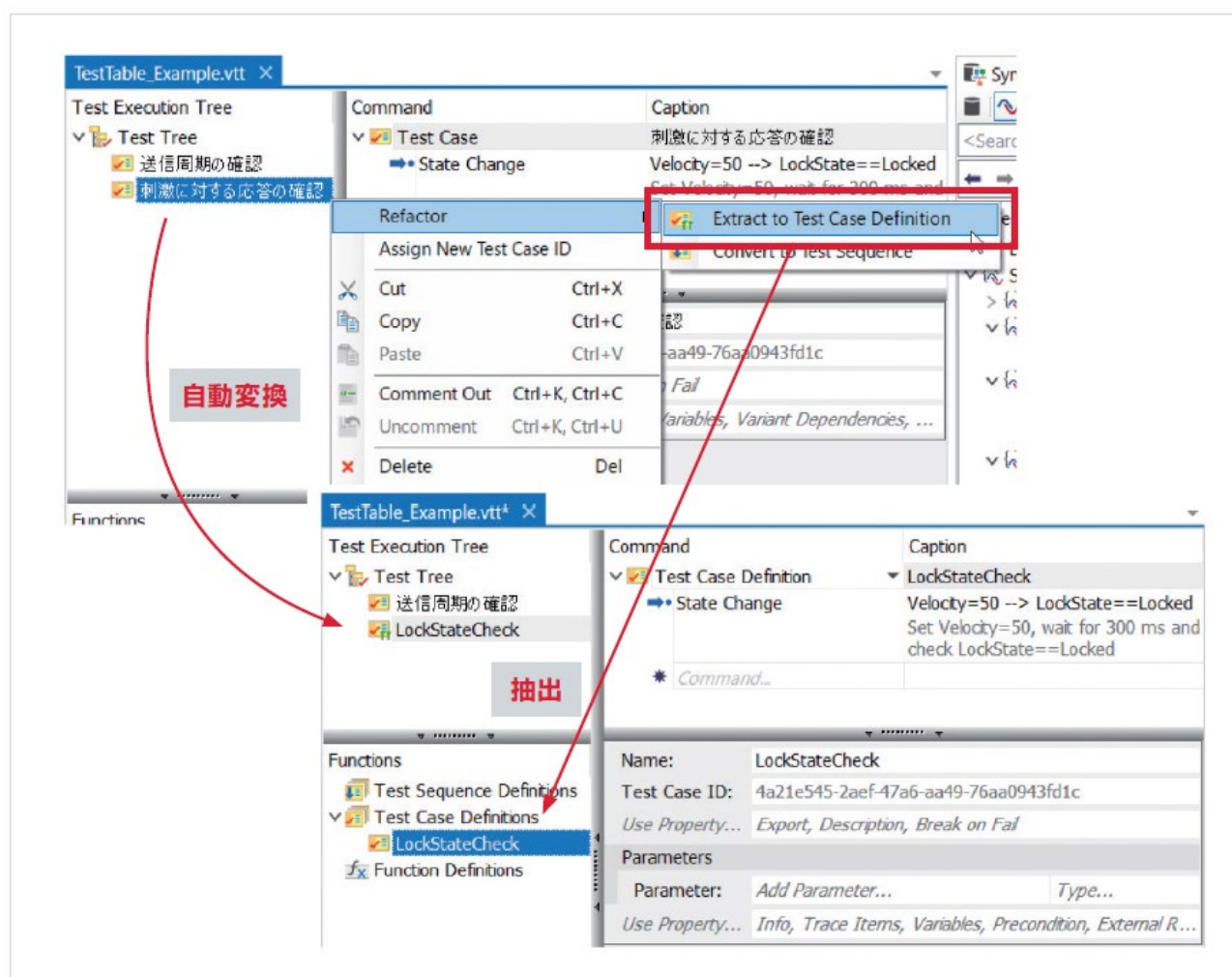


図 33: 既存のテストケースから Test Case Definition を抽出

※Test Case Definition が生成され、元のテストケースはこの呼び出しに変換される

この機能を使用すれば、作成済みのテストケースを容易に Test Case Definition に変換することができます。

■ テストケースリスト

そして、vTESTstudio では前記に加えて、テストケースリストという機能により、「Test Case Definition」をリスト形式で一度に大量に呼び出すことができます。

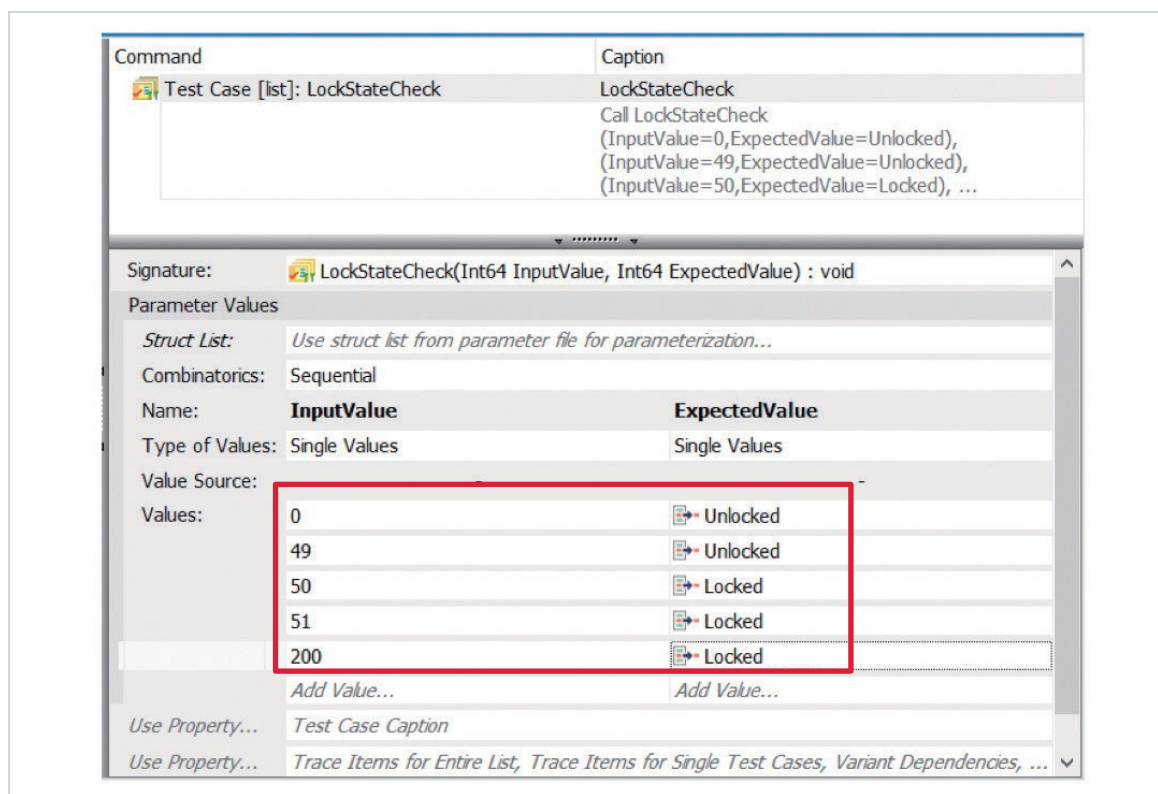


図 34: テストケースリスト機能を利用することで、Test Case Definition をリスト形式で一氣に呼び出し

図 34 の例では、5 つのテストケースをリスト形式で一度に生成しています。このように、Test Case Definition とテストケースリストを使用することで、同じパターンのテストケースを容易に、かつ、網羅的に作成することができます。

■ シンボルを引数に利用し、より柔軟にテストを再利用

Test Case Definition では引数に数値だけでなくシンボルも指定できます。「刺激に対する応答」を確認するテストは通常、特定のシグナルのペアだけに適用するものではなく、多くのシグナルや I/O の組み合わせに対して実施します。そのため Test Case Definition はテスト対象のシンボルも引数として設定することができ、同じパターンのテストに対してテスト対象のシンボルを柔軟に変更可能となっています。

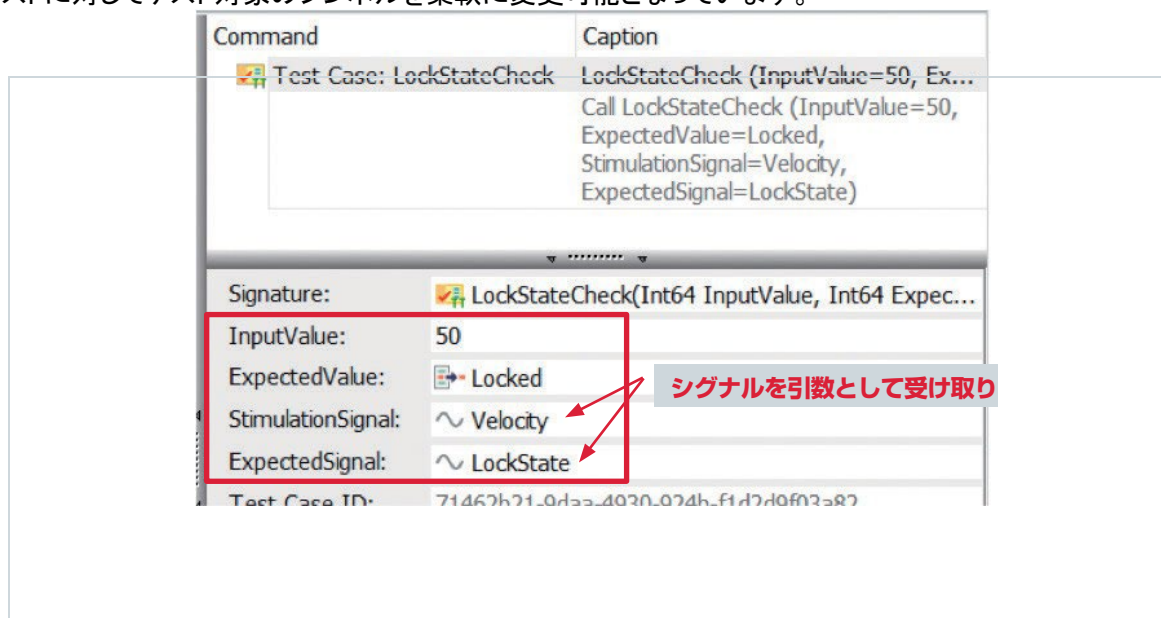


図 35: Test Case Definition はシンボルも引数として設定できるため、同じテストのパターンを他のシグナル等にも適用可能

2.6.2 Parameter Editor の使用

vTESTstudio の補助的なエディターである Parameter Editor を使用することで、テストの再利用は更に効率化します。Parameter Editor は、テスト用の定数を定義するパラメーターファイルについて作成・編集するエディターです。パラメーターファイルには通常の定数を定義するだけでなく、リストや構造体、構造体リストの定義も可能です。テストで使うさまざまな値をテストケースから分離し、パラメーターファイルに一元管理することで、後から値変更や値の追加が必要になった際の工数を大きく削減できます。

The screenshot displays the vTESTstudio interface with the Parameter Editor open. It shows three main sections: a table of scalar parameters, a list of values, and a structure list. A red arrow points from the 'MonitoringTime' parameter in the first table to its usage in a test case command.

Name	Value
LockVelocity [km/h]	50
MaxResponseTime [ms]	500
MinResponseTime [ms]	100
MonitoringTime [ms]	5000

スカラー値のパラメーター

Name	Values
VelocityList [km/h]	[0] 10 [1] 25 [2] 50 [3] 100

リスト

LockState Test		
Velocity	LockState	
0	Unlocked	
49	Unlocked	
50	Locked	
51	Locked	
200	Locked	

構造体リスト

Command	Caption
Test Case	送信周期の確認
Cycle Time Relative	CentralLockingSystemState [0.9..1.1] Check relative cycle time of CentralLockingSystemState to be within [0.9 .. 1.1]
Wait	MonitoringTime ms Wait for MonitoringTime ms
Comm	

パラメーターをテストケースに使用

Time: MonitoringTime ms
Caption: MonitoringTime ms

図 36: パラメーターファイルを使用することで、テストシナリオと使用する値(テストデータ)を分離。テストケースの再利用性を高める

■ パラメーターファイルの構造体リストをテストケースリストで利用

前述の 2.6.1 項の例では Test Case Definition とテストケースリストの使用により、テストの再利用を行いました。その際、テストで使用する値は、テストケースリストの設定画面上で直接入力しました。これに対し、値をパラメーターファイル上で構造体リストとして定義し、テストケースリストに適用することで、テストの再利用性が更に向上します。

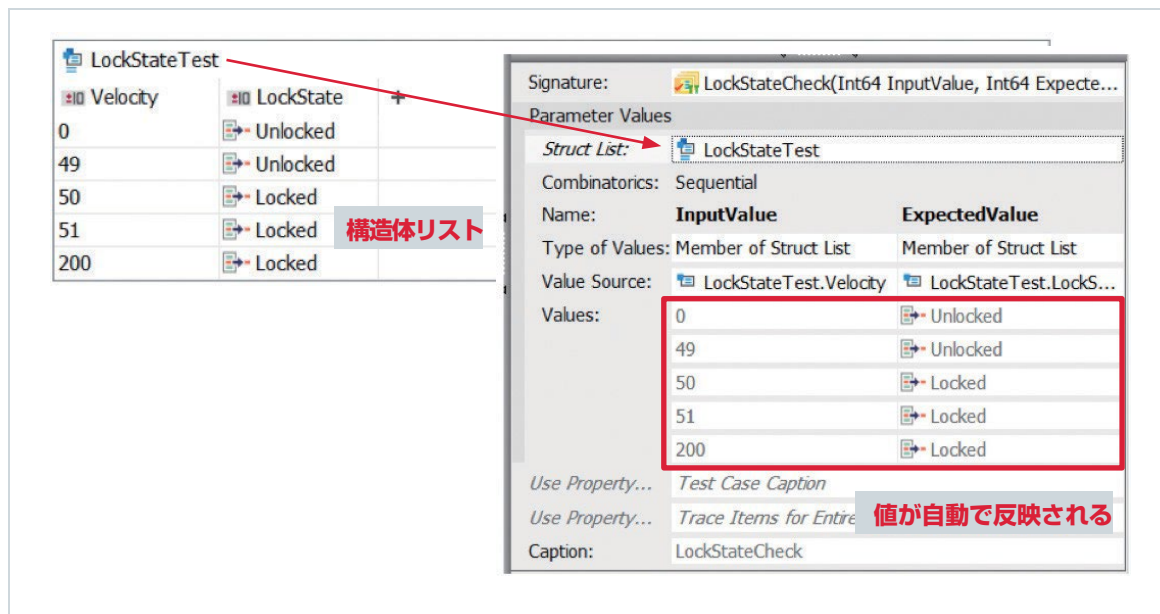


図 37: 構造体リストをテストケースリストのデータとしてそのまま利用可能 ※リストやスカラー値のパラメーターもちろん使用可能

パラメーターファイルは MS Excel®でも編集可能です。また、vTESTstudio の設定により、CANoe でテスト実行直前に(vTESTstudio 上の変更や、再度のビルド作業なしに)パラメーターだけを MS Excel®で変更する、という利用方法も可能です。特に、テストケースリストと構造体リストのパラメーターが紐づいている場合、構造体リストの行を追加・削除することで、CANoe でテスト実行直前にテストケースの数を増減させることもできます。

	A	B	C	D	E
30	StructName	LockingTests::LockStateTest			
31	ParameterName		Velocity	LockState	
32	Type		Integer	Integer	
33	Info				
34	Values			0 VtSig_CAN::SUT::CentralLockingSystemState::LockState::Unlocked	
35				49 VtSig_CAN::SUT::CentralLockingSystemState::LockState::Unlocked	
36				50 VtSig_CAN::SUT::CentralLockingSystemState::LockState::Locked	
37				51 VtSig_CAN::SUT::CentralLockingSystemState::LockState::Locked	
38				200 VtSig_CAN::SUT::CentralLockingSystemState::LockState::Locked	
39					

図 38: パラメーターファイルの実体は CSV ファイルのため、MS Excel®での編集が可能

このように、テストのシナリオを Test Table Editor で作成し、テストで使用する値をパラメーターファイルで一元管理すると、パラメーターの変更だけで、作成済みのテストを容易に再利用できるようになります。

なお、上記のようにテスト用のデータをスプレッドシートなどで管理し、テストのシナリオと分離して再利用する方法を、アカデミックな用語で「データ駆動テスト」と呼びます。

2.6.3 テストケースのパーツを再利用する

前項までの例ではテストケースのひな型を作り、これに与える値をパラメーターファイルで定義することでテストの再利用を行いました。本項では、より柔軟にテストケースのパーツについて再利用する方法を紹介します。

テストケースのパーツとは、いくつかのテストステップの集合体、という意味になります(テストステップ 1 つの場合も含まれます)。前項までで紹介した Test Case Definition はテストケースに対するひな型を作成しましたが、テストケースのパーツに対しては Test Table Editor の

「Function Definition」を使用します。これにより、よく使用する一連の処理をまとめ、分かりやすい名前を付けることで容易に再利用できるようになります。図 39 はその一例となります。

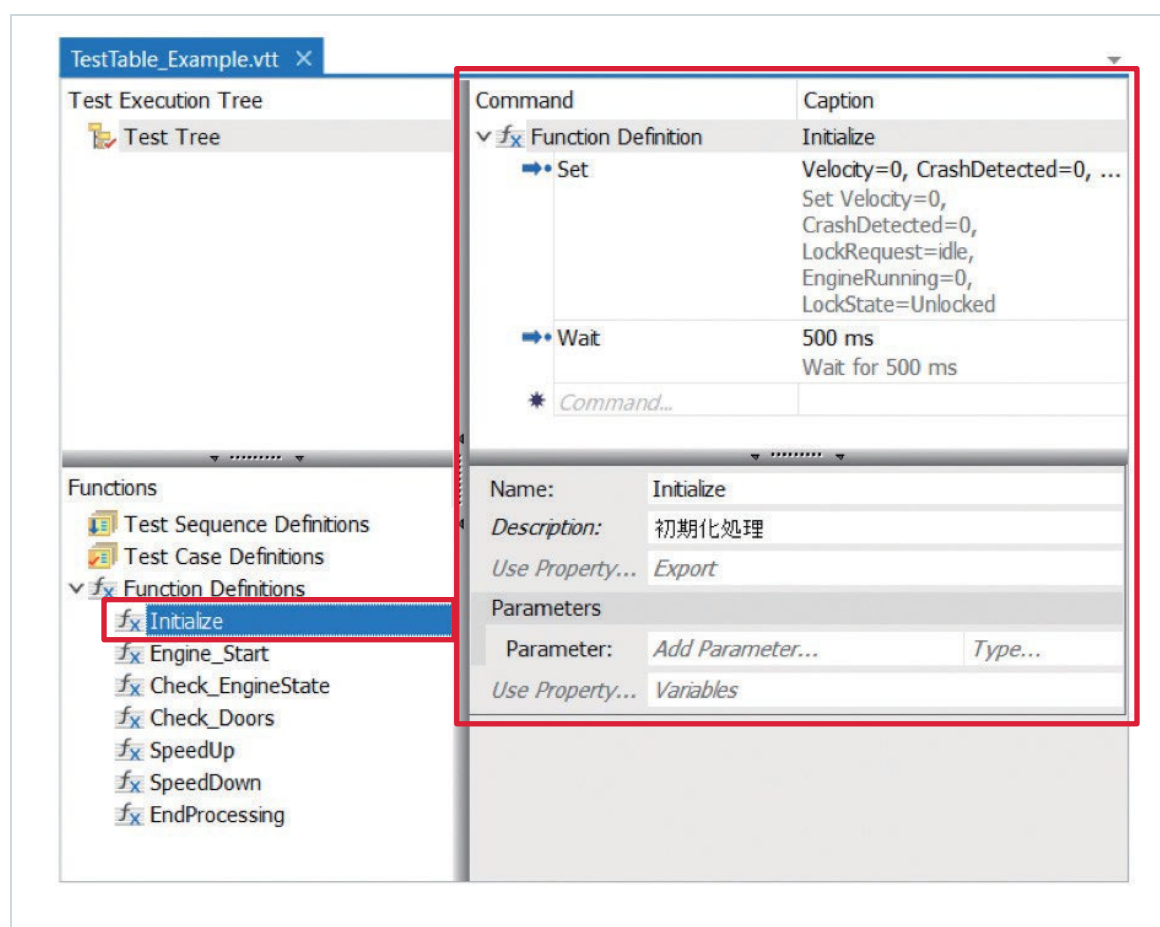


図 39: よく使用する一連の処理は Function Definition に定義することで再利用性が向上

Function Definition も Test Case Definition と同様、引数を設定し、値を外部から設定できます。

Function Definition 領域に定義した Function は、テストケースから呼び出して使用できます。場合によっては、以下のように Function を並べるだけでテストが実装できることもあります。また、Function から Function を呼び出すことや、Test Case Definition から Function を呼び出すことも可能です。

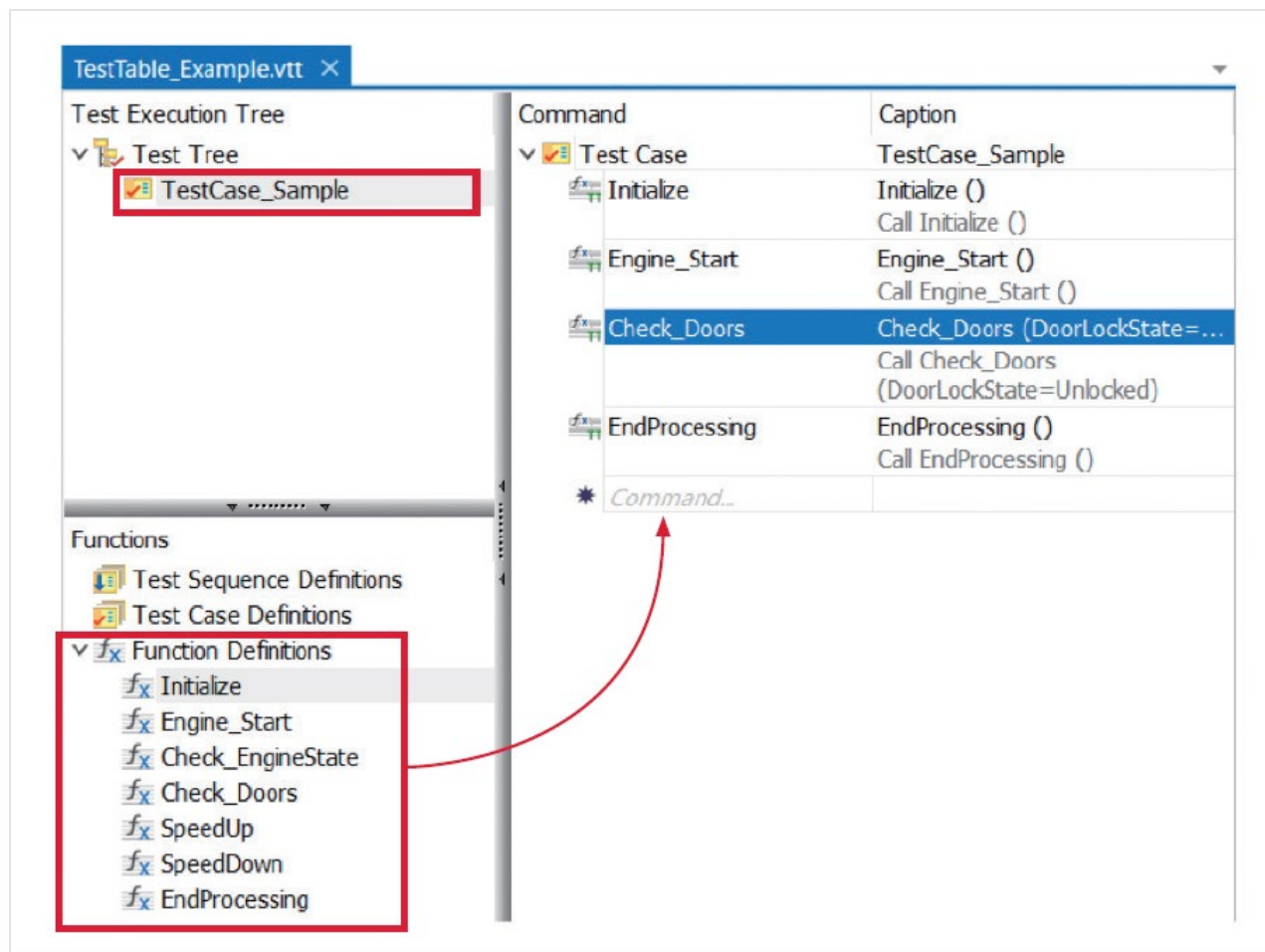


図 40: テストケースから Function を呼び出して使用。この例では Function を呼び出すだけでテストケースの実装が完了

ここで見たように、テストで行ういくつかの処理を Function のような形でまとめ、これらを組み合わせてテストを実装していく方法を、アカデミックな用語で「キーワード駆動テスト」と呼びます。この場合、定義した Function が「キーワード」に該当します。

■ Function の抽出

テストケースの抽出と同様に、テストケース内の任意のパーツを選択し、Function Definition へ変換する機能もあります。

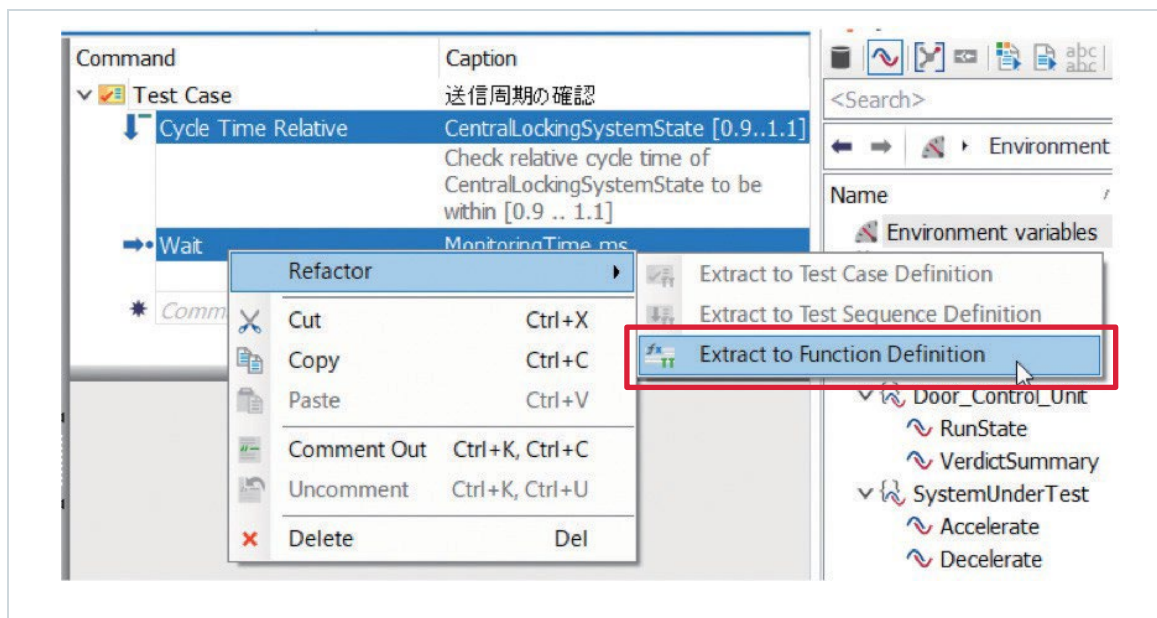
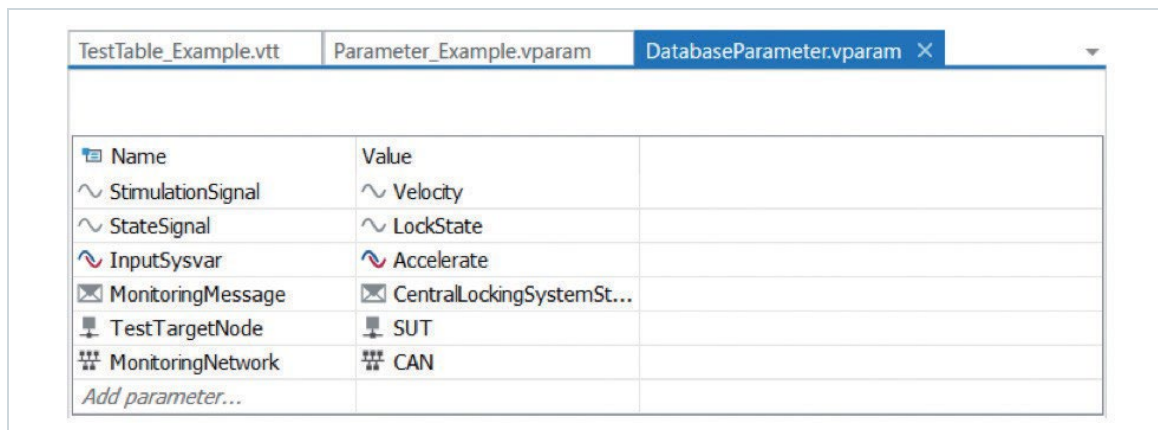


図 41: テストケース内の対象とする処理を選択し、Function に変換

この機能を使用すれば、作成済みのテストケースから容易に Function を作り出すことができます。

2.6.4 テスト対象のシグナルやメッセージの汎用化

テストの再利用に関し、最後にデータベースに依存しないテスト実装について簡単に紹介します。前述のパラメーターファイルには、数値や文字列などの値だけでなく、シグナルやシステム変数、メッセージやノードなども定義可能です。この機能を使用すると、たとえばメッセージの送信周期を確認するテストのテスト対象メッセージをパラメーター化し、容易に他のメッセージに置き換えることができます。



Name	Value
StimulationSignal	Velocity
StateSignal	LockState
InputSysvar	Accelerate
MonitoringMessage	CentralLockingSystemSt...
TestTargetNode	SUT
MonitoringNetwork	CAN
Add parameter...	

図 42: パラメーターファイルにはメッセージやシグナル、システム変数などのシンボルも定義可能

この機能を応用することで、データベースに依存しないテストを実装できるようになります。データベースに依存しないテストを実装する場合、パラメーターファイルの編集工数が増えることや、テストケースをデータベースに依存しない形で作り込む必要があるなど、工数が増える箇所もあります。一方、他のデータベースに対してテストを流用する際、テストケースの変更箇所を大幅に減らすことができるようになります。そのため、どのような ECU にも実施する汎用的なテストについては、大きなメリットが得られます。

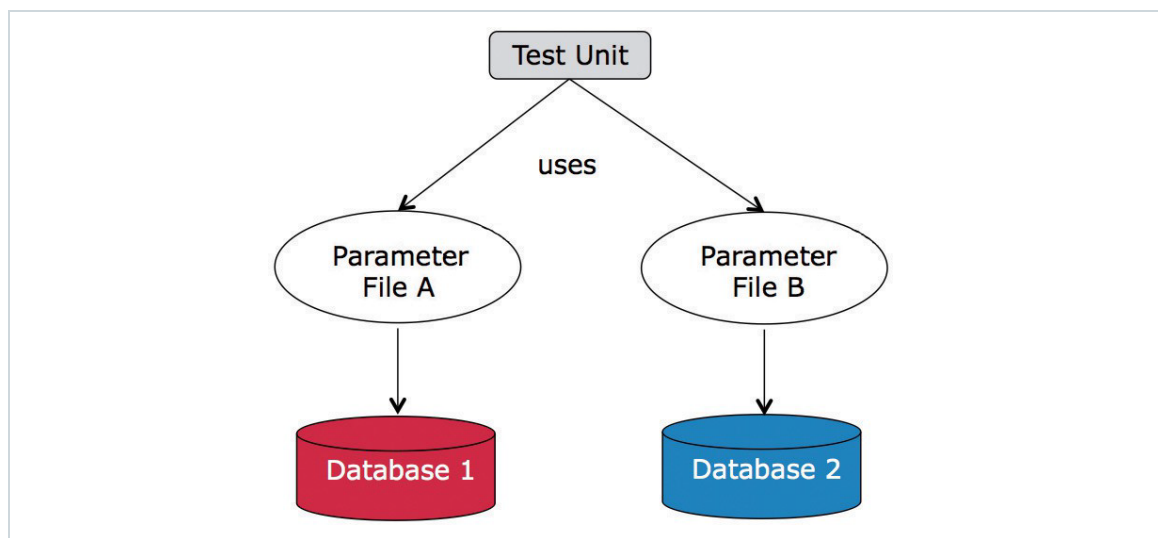


図 43: パラメーターファイルを利用し、テストケースからデータベースの依存性を排除した構成例

本節の冒頭で述べたとおり、テスト自動化において、テストの再利用は非常に重要です。そのため本節では、vTESTstudio の持つテスト再利用に関する機能について、かなりのページを割いて紹介しました。本節の内容や、その他の再利用に関する機能の詳しい操作方法については、vTESTstudio のアドバンスマニュアルを参照してください。

 01.アドバンスマニュアル	--	06.08.2020 06:40	30.09.2020 05:30	補足やTIPSとなるマニュアル（各種設定方法、プログラミング、CANdbの取り扱い等）を掲載しています
---	----	---------------------	---------------------	---

2.7 固有の仕様に基づく複雑なテストを作成

これまで見てきたように、Test Table Editor にはさまざまな定義済みのコマンドがあり、ECU のテストを実装する上で十分な機能を提供しています。しかし、本稿 2.1 節の表 E の No.8 に記載したように、ECU 固有の仕様に関するテストや、込み入った複雑な処理を必要とするテストを行う場合には、定義済みコマンドだけでは対応できないケースもあります。

vTESTstudio はこういった場合についても容易に対処できるよう、Test Table Editor と Coding Editor を連携させる機能を提供しています。

以下は vTESTstudio の Coding Editor で CAPL によるテスト関数を作成している例です。Coding Editor では CAPL または C#を用いてテストケースやテストステップを作成できます。

```

12 export testfunction SampleTestFunction()
13 {
14     testWaitForTimeout(1000);
15     testStepPass("Verdict", "Sample Test Step");
16 }

```

図 44: CAPL で作成したテスト関数の例。あくまで例であるため、ここでは単に 1000ms の待機とレポートへの出力のみを実装

Coding Editor でテストケース全体を実装すると、多くの場合、工数が大きくなります。そのため、定義済みコマンドにはない固有の処理や、複雑な処理のみをユーザー定義のテストコマンドとして Coding Editor で実装し、テストケース全体は Test Table Editor などを実装すると効果的です。

上記のテスト関数の例では関数の宣言文の先頭に「export」キーワードが配置されています。このキーワードがある関数は、下記の図のように vTESTstudio 上の「User Functions」というビューに表示されます。ここに表示された関数は、ユーザー定義のテストコマンドとして、Test Table Editor などの他のエディターから定義済みコマンドと同じように使用できます。

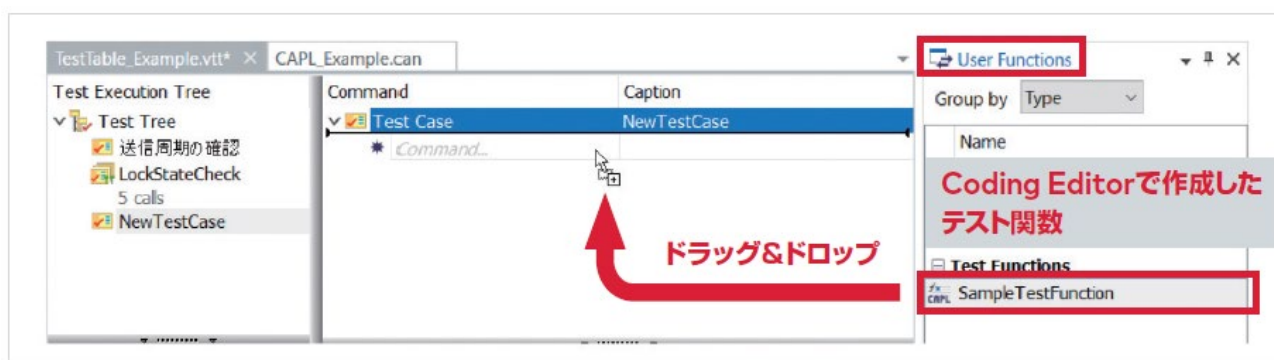


図 45: CAPL で作成したテスト関数をユーザー定義のコマンドとして Test Table Editor で使用

一度、専用のテスト関数を実装してしまえば、以降はライブラリーとして別のプロジェクトに再利用することも容易です。このように vTESTstudio では汎用的なテストだけでなく、複雑な ECU 固有の仕様などに対しても効率的に実装できるようになっています。

Test Table Editor と Coding Editor の連携に関する詳しい操作方法については、vTESTstudio のアドバンスマニュアルを参照してください。

 01.アドバンスマニュアル	--	06.08.2020 06:40	30.09.2020 05:30	補足やTIPSとなるマニュアル（各種設定方法、プログラミング、CANdbの取り扱い等）を掲載しています
---	----	---------------------	---------------------	---

2.8 波形を定義して ECU への刺激として入力

本稿 2.5 節では、刺激に対する応答のテストについて単純な例を説明しましたが、車速の時間変化に対する応答確認など、一つのテストケース内で何度も刺激を与え、それぞれに対する応答を確認するテストも広く実施されています。こういったテストの場合にも、2.5 節に記載の方法を繰り返すことで実装可能ではありますが、本稿 2.1 節の表 E、No.9 に記載のように、波形を描く形で ECU への刺激を定義する方が視覚的に分かりやすくなります。

vTESTstudio には波形を定義するための専用エディターとして、Waveform Editor があります。

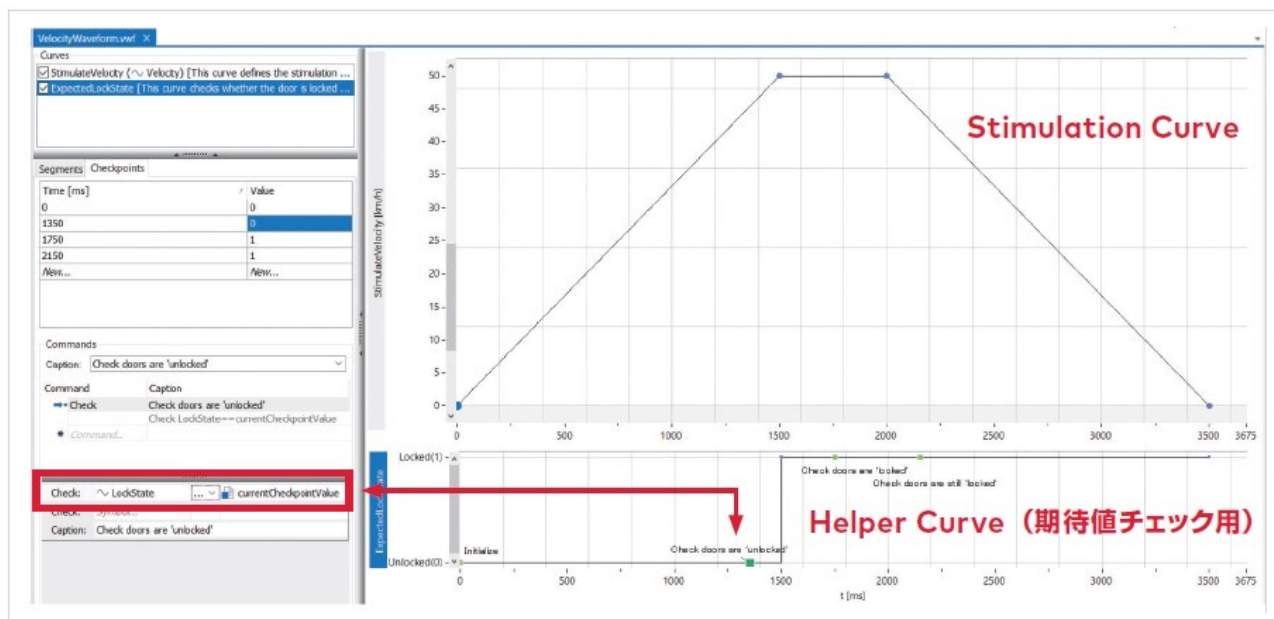


図 46: Waveform Editor で刺激用の波形を定義。波形にコマンドを挿入するためのチェックポイントも定義可能

Waveform Editor はテスト設計エディターではなく、補助的なエディターであるため、直接テストケースを作成できるわけではありません。Waveform Editor で波形を定義した後で、この波形を Test Table Editor など呼び出して使用します。呼び出す際に、波形を適用するシグナルやシステム変数を割り当てます。

■ Waveform Editor 上でコマンドを使用

Waveform Editor では、図 46 のように波形にチェックポイントを定義し、値チェックのためのコマンドを使用できるようになっています。そのため、Test Table Editor 上のテストケースでは波形を呼び出すだけとし、値の刺激と応答期待値のチェックをすべて Waveform Editor 上に定義して、実質的に Waveform Editor でテストケースを実装する、というような使い方も可能です。一方、このような使い方をすると、定義する波形の種類やコマンドの数が多くなりがちで、視覚的な分かりやすさが失われて実装効率が落ちる懸念があるため、Waveform Editor 上のコマンドの使用には注意が必要です。

Waveform Editor の詳しい操作方法については、vTESTstudio のアドバンスマニュアルを参照してください。

 01.アドバンスマニュアル	--	06.08.2020 06:40	30.09.2020 05:30	補足やTIPSとなるマニュアル（各種設定方法、プログラミング、CANdbの取り扱い等）を掲載しています
---	----	---------------------	---------------------	---

2.9 テスト全体を一覧表示し、テストの抜け漏れチェック

Test Table Editor は標準的なエディターであり、さまざまなテストに対応できます。一方、テストケースを一つ一つ個別に実装するタイプのエディターであるため、テスト再利用の機能を使用したとしても、テストケースの数や種類が多くなってくると全体像が見えづらくなります。

本稿 2.1 節の表 E、No.10 に記載のように、テスト全体を一覧表示するには、Test Table Editor ではなく Test Sequence Diagram Editor を使用すると効果的です。ただし、このエディターの使用にはライセンスオプションの「Graphical Test Design」が必要になります。

Test Sequence Diagram Editor では、以下の図のようにグラフィカルにテスト全体を実装します。

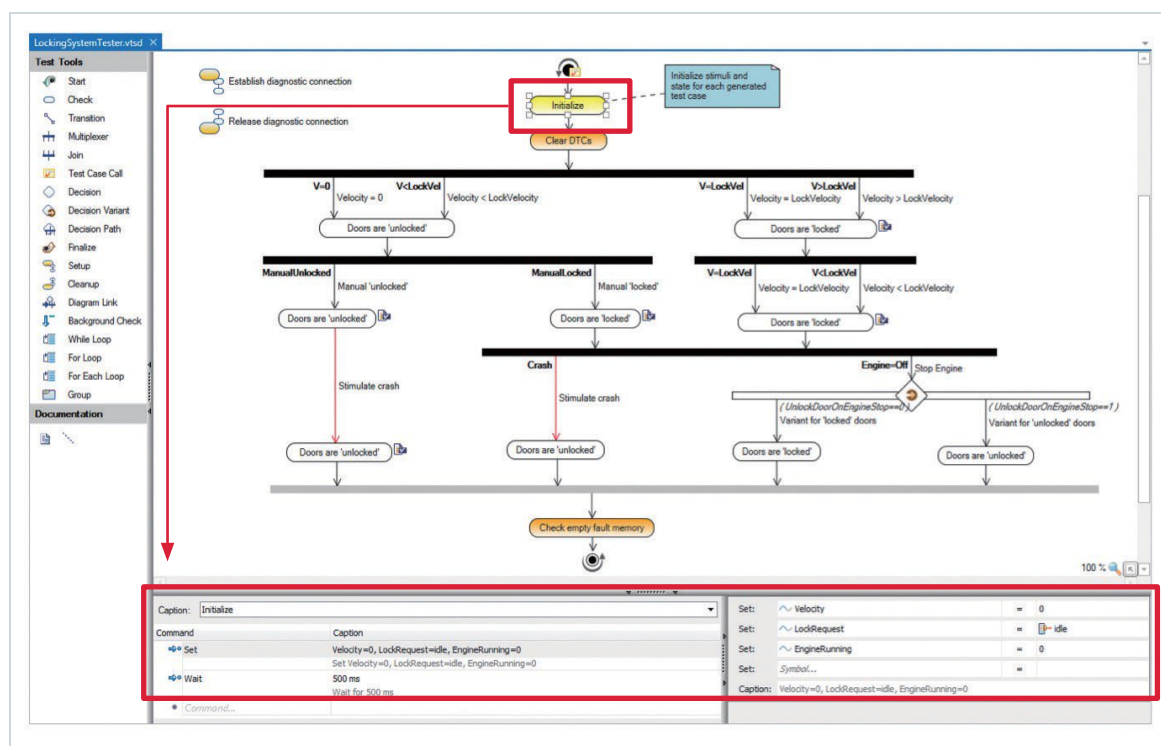


図 47: Test Sequence Diagram Editor はテストのロジックをダイアグラムによりわかりやすく表示。エレメント内部は下段で実装

ここで描画されているダイアグラムのエレメントには、Test Table Editor で使用可能な定義済みコマンドや、Function Definition で定義した Function、Coding Editor で作成した関数などがそのまま使用できます。図 47 の例では、開始を示すアイコン「」の直下にある黄色の「Initialize」エレメントの内部を、下段で実装しています。

Test Sequence Diagram Editor は、描画されたダイアグラムの開始位置「」から終了位置「」までの取り得るすべての経路を計算し、一つの経路を一つのテストケースとして生成します（※設定により、別の使用方法も可能です）。

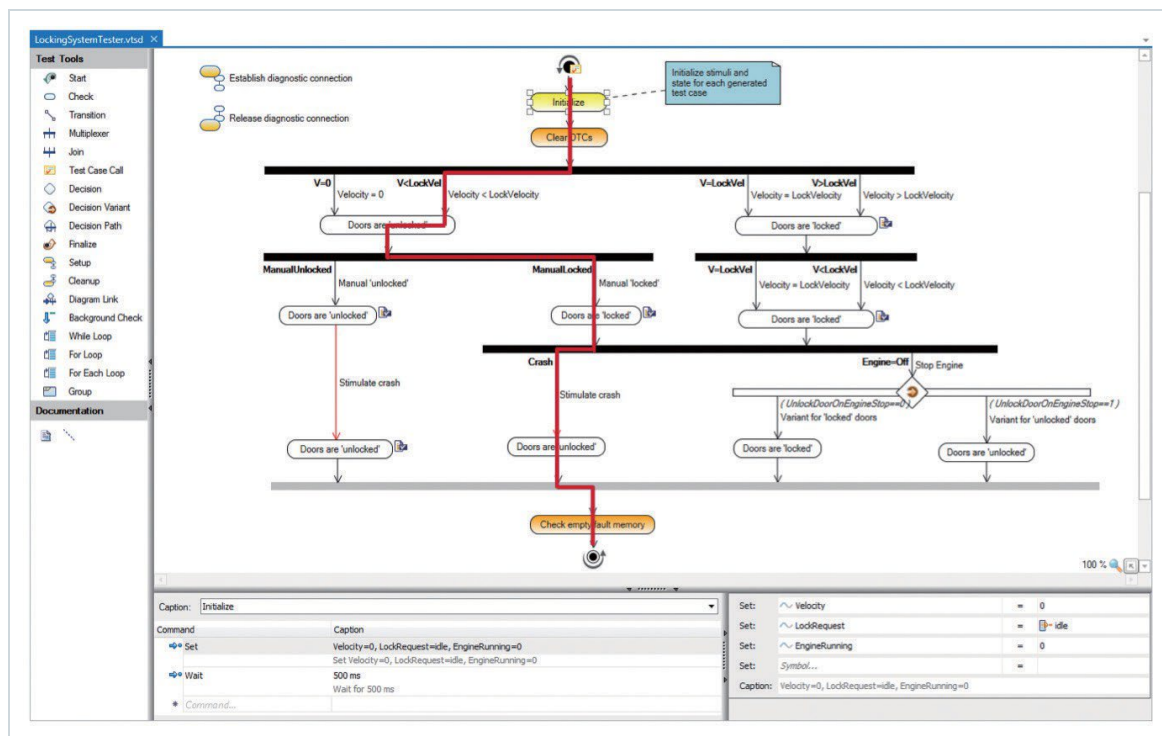


図 48: Test Sequence Diagram Editor は取り得るすべての経路を自動計算し、テストケースを生成

たとえば図 48 のように、赤い矢印で辿っている経路が 1 つのテストケースとなります。

つまり、ダイアグラム全体と個々のエレメントの内部処理を実装すると、すべてのテストケースをエディターが生成してくれます。

また、エディターが生成したテストケースの個々の内容を確認するために、Path View という機能も提供されています。この機能を使用することで、生成するテストケースがいくつあるのか、またどのような処理が行われるのか、一覧できるようになります。

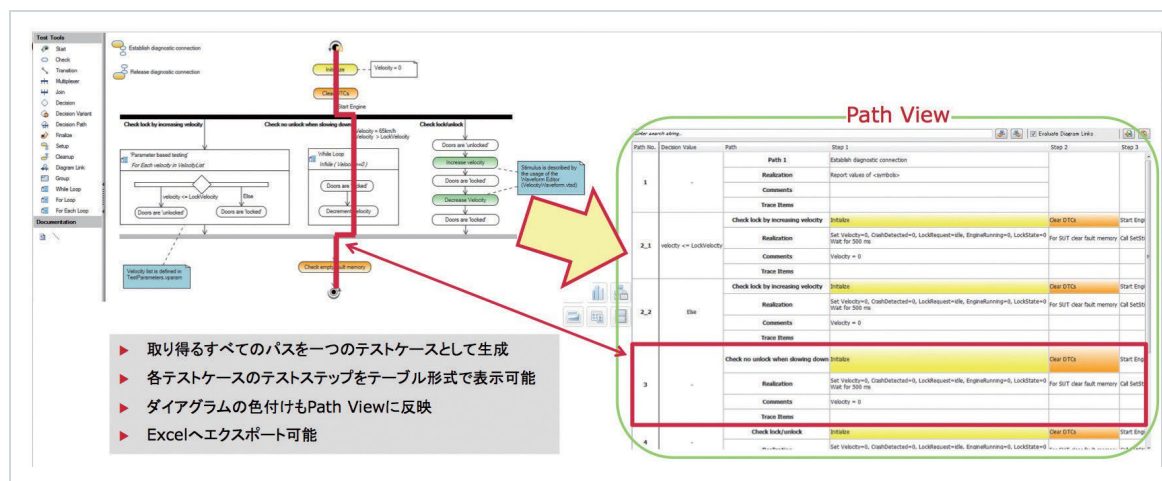


図 49: Path View 機能により、生成されるテストケースの処理を一覧できる

■ Test Sequence Diagram Editor の利点

このエディターの利点を簡単にまとめると以下のようになります。

- > エレメントに実装した処理を容易に他のテストケースに再利用できること
- > ダイアグラムの描画とエレメントの実装について、それぞれ担当者を分けることで、複数人で同時に実装できること
- > エレメントの実装について、Test Table Editor や Coding Editor で実装した処理を流用できること
- > テストの流れが視覚的に分かりやすくなること
- > すべてのテストケースを一覧表示し、抜け漏れがないか分かりやすく確認できること

特に最後の項目は、このエディターがテスト実装だけでなく、実装したテストのレビューにも活用できることを示しています。

本節では Test Sequence Diagram Editor について簡単に紹介しました。Test Sequence Diagram Editor の詳しい操作方法や、その他の機能については、vTESTstudio のアドバンスマニュアルを参照してください。

01.アドバンスマニュアル	--	06.08.2020 06:40	30.09.2020 05:30	補足やTIPSとなるマニュアル（各種設定方法、プログラミング、CANdbの取り扱い等）を掲載しています
---------------	----	---------------------	---------------------	---

2.10 その他の主な機能

vTESTstudio には多様なエディターや豊富な機能があるため、本稿だけでこれらすべてを紹介することはできません。本節では、ここまでで紹介しきれなかったその他の主要な機能について、簡単に概要のみ説明します。これらのエディターや機能の詳細にご興味のある方は、vTESTstudio の製品資料、またはアドバンスマニュアルを参照してください。

> その他のエディター

エディター	内 容
State Diagram Editor	State Diagram Editorは状態遷移図を用いてテストを実装するエディターで、使用にはオプション「Graphical Test Design」が必要です。なお、製品設計に使用する状態遷移図からテストを生成するエディターではなく、テスト設計・実装の手段として状態遷移図を用いるエディターであるため、注意が必要です。
Classification Tree Editor	テストベクターを作成するための補助的なエディターです。同値分割や境界値分析についてツリー構造を使用して視覚的に表現し、値の組み合わせを自動生成する機能を持っています。

■ その他の主な機能

機 能	内 容
vTESTstudio Viewer	vTESTstudio Viewerは無償配布のアプリケーションで、vTESTstudioのライセンスをお持ちでない方でも、vTESTstudioで実装されたテスト内容を確認できます。vTESTstudioと比べ、編集機能等がないだけで、他の機能はすべて同一です。 vTESTstudio ViewerはCANoeインストール時に同時にインストールされます。またベクターのWebサイトからダウンロードすることも可能です。
バリエーション機能	vTESTstudioにはテストのバリエーションを設定する機能があります。たとえばECUの仕向け先やグレードなどの違いにより、使用するパラメータの値や実行するテストケースが異なる場合に、全く別のvTESTstudioプロジェクトを作成しなくとも、バリエーションを設定しておくことで、これらの差分を吸収できるようになります。
トレーサビリティ	vTESTstudioは管理ツールと連携し、管理ツールにて生成された要件やテスト仕様と、vTESTstudio上のテストケースをリンクさせることができます。また、CANoeでテスト実行して生成されるレポートにも自動的にこのリンクが引き継がれ、レポートを管理ツールへインポートした際に、テストの結果が要件やテスト仕様と紐づけされます。これにより、管理ツール→vTESTstudio→CANoe→管理ツールと、テストのトレーサビリティを確保することができます。 連携できる管理ツールにはベクター製のPREEvisionやIBM Doors、PTC Integrityなど多岐にわたります。
ファジング機能	ファジングテストとは、テスト対象の堅牢性を確認するためのテストで、セキュリティ上の脆弱性を発見する用途にも使用されます。多くの場合、ミスの起こりやすいデータやランダムなデータなどをテスト対象に与え、その応答を確認するテストとなります。vTESTstudioはこのファジングテストに関する機能を有しており、容易にファジングテストを実装できます。

3. よくあるお問い合わせ

3.1 vTESTstudio の操作方法を詳しく知りたい

ベクターでは以下のようなコンテンツを準備しておりますので、vTESTstudio を実際にお使いになる際にはご活用ください。

- > vTESTstudio トレーニング
- > vTESTstudio に関するアドバンストマニュアル
- > vTESTstudio の自習用サンプル(「CANoe テスト機能、vTESTstudio 自習用サンプル」)

トレーニングはベクターにて開催しており、以下 Web サイトから内容や開催日程を確認できます。

https://vector-academy.com/vj_training_jp.html

アドバンストマニュアルや自習用サンプルは、ベクター製品ユーザー向けコンテンツを提供している「ベクター・ジャパン Member Area」に掲載しています。

詳細は下記リンク先をご覧ください。(ベクターの Web サイトへ移動します)

<https://www.vector.com/jp/ja/support-downloads/vj-memberarea/>

3.2 CANoe など vTESTstudio 以外のツールの使い方が知りたい

本稿で紹介したテストソリューションに関するツールの詳細については、以下の資料などを参照してください。なお、これらの資料はすべてベクターの Web サイトからダウンロードできます。

製 品	参考資料
CANoe	はじめてのCANoe、アドバンストマニュアル「CANoe インタラクショナルレイヤー送信モデルの作成」など
VTシステム	アドバンストマニュアル「V Tシステム ユーザーズマニュアル」など
Test Report Viewer	アドバンストマニュアル「CANoe Test Report Viewer 基本操作」など

3.3 CANoe でもテスト実装できますが、vTESTstudio を使用するメリットは何ですか？

vTESTstudio を使用するメリットは、CANoe 単体でのテスト実装に比べて、工数を大幅に削減できることです。

自動テストにおいては、テスト実装が唯一工数のかかる工程となる上、必要工数が大きいため、テスト実装の効率化が非常に重要です。vTESTstudio はテスト実装を効率化するために作られたツールであり、効率化のためのさまざまな機能を持っています。

CANoe 単体でのテスト実装と vTESTstudio を使用した実装との違いについては、本稿 1.1.2 項と 1.2 節を参照してください。また vTESTstudio の持つさまざまなテスト実装効率化機能については、本稿 2 章を参照してください。

3.4 実際に導入した事例について知りたい

弊社のお客様からいただいた事例をまとめた「Vector Case Study Book - ECU テストソリューション」にて、さまざまな ECU テストにおける導入事例を紹介しております。弊社 Web サイトの以下の URL からダウンロードできますので、参照してください。

https://assets.vector.com/cms/content/know-how/VJ/PDF/VectorCaseStudyBook_TestSolution.pdf

3.5 テスト自動化をどこから、どのように進めたらよいのか分かりません

自動テストは一般的に、テスト実装に工数が必要な一方、テストの実行には工数がかからないため、繰り返し行うテストが向いています。そのため、いきなりすべての手動テストを自動化するのではなく、最初は多くの ECU に実施する、汎用的で良く利用されるテストのみ自動化することをお勧めします。その後、少しずつ自動テストを増やしていくことで、比較的少ない負担で、効果を確認しながら自動化の範囲を拡げていくことができます。また、必ずしもすべての手動テストを自動化する必要はありません。

具体的には、本稿 2.3 節や 2.4 節、2.5 節でご紹介しているような、CAN 通信や診断通信、単純な刺激に対する応答のテストなどは、汎用的で繰り返し実施されるテストに該当します。これらのテストから自動化をはじめ、これらのテストを再利用しながら、徐々に複雑なテストを自動化していくことをお勧めします。

3.6 CAN 以外のプロトコルについてもテスト実装できますか？

はい、できます。CANoe のオプション製品である CANoe.LIN や CANoe.Ethernet などを使用することで、CAN 以外のプロトコルでも CANoe 上で解析・シミュレーション・テストの機能が使用できるようになります。vTESTstudio 自体には特にプロトコルに関するオプションや制約等はなく、テストはこれら CANoe のオプションと関係なく実装可能ではありますが、実際にはテスト実行環境が必要となるため、上記の CANoe のオプションが必要となります。また、ベクターテストソリューションには、LIN コンフォーマンステストや Open Alliance

TC8(車載 Ethernet ECU のコンフォーマンステスト)といった、プロトコル固有の規格化されたテストを自動実行するソリューションもあります。詳しくは弊社ベクター・ジャパン Member Area に掲載の「製品説明スライド」などを参照してください。

3.7 ダイアグラムエディターはオプションですか？

はい。Test Sequence Diagram Editor の使用には vTESTstudio のオプションである「Graphical Test Design」が必要です。なお、このオプションをお持ちの場合は State Diagram Editor もご利用可能です。

3.8 vTESTstudio の実装内容について vTESTstudio を持っていない人にレビューしたい

vTESTstudio Viewer という無償配布のアプリケーションを使用することで、vTESTstudio のライセンスをお持ちでない方でも vTESTstudio で実装されたテストの内容を確認できます。vTESTstudio Viewer は vTESTstudio と比べ、保存・編集機能がないだけであり、他の機能はすべて同一です。

vTESTstudio Viewer は CANoe インストール時に同時にインストールされます。またベクターの Web サイトからダウンロードすることも可能です。

また、vTESTstudio Viewer を使用する以外の方法として、vTESTstudio は実装内容を PDF ファイルへエクスポート可能ですので、PDF 化してレビューする方法もあります。

3.9 ECU 内部値のテストはできますか？

はい、できます。CANoe のオプション製品である CANoe.AMD/XCP を使うと、内部値の情報を定義した A2L ファイルからシステム変数を自動生成できます。このシステム変数を使用すれば、vTESTstudio で容易に内部値のテストを実装できます。本稿の 2.5 節でも内部値に関するテストについて記載していますので、参照してください。

3.10 VT システムと vTESTstudio との連携はできますか？

はい、できます。CANoe 上で VT システムの設定を行うと、VT システム制御用のシステム変数が自動生成されます。このシステム変数を使用すれば、vTESTstudio で容易に VT システムを制御するテストを実装できます。本稿の 2.5 節でも VT システムに関するテストについて記載していますので、参照してください。

またこの他、vTESTstudio には VT システム設定用の専用コマンドが用意されています。こちらは vTESTstudio のアドバンスドマニュアルや、「CANoe テスト機能、vTESTstudio 自習用サンプル」を参照してください。

3.11 Simulink などのモデルに対してテストを行うことはできますか？

はい、できます。CANoe には Simulink と連携するための専用インターフェイスがあり、これを使用することでモデルをテスト対象とすることができます。詳しくは CANoe - Simulink 連携に関するアドバンスドマニュアルを参照してください。

3.12 実装中の ECU ソフトウェアに対してテストを行うことはできますか？

はい、できます。CANoe には vVIRTUALtarget という製品があり、開発中の AUTOSAR ECU のソフトウェアを CANoe 上で仮想的に動作させることができます。vVIRTUALtarget を使用することで、ECU ソフトウェアに対するテストも可能です。詳しくは vVIRTUALtarget に関する製品資料を参照してください。

あとがき

以上、ベクターテストソリューションにおける vTESTstudio の役割、および vTESTstudio で実現できることについて解説しましたが、いかがでしたでしょうか。本稿『はじめての vTESTstudio』が vTESTstudio についてのすべてを網羅しているわけではありませんが、vTESTstudio の基本的な利点をご理解いただけたのではないかと思います。vTESTstudio についてさらに詳細をお知りになりたい方は、ベクターWeb サイトにて「アドバンスマニュアル」、「CANoe テスト機能、vTESTstudio 自習用サンプル」、「サンプルコンフィギュレーション」など各種有用なドキュメントをご用意しておりますので、是非、ご覧ください。また、ベクターでは vTESTstudio に関するトレーニングも開催しておりますのでこちらもご活用ください。本稿が、車載ネットワーク開発に携わるエンジニアの方々の一助となれば幸いです。

4. お問い合わせ窓口

技術関連のお問い合わせは、ベクターサポートまでお寄せください。

- ✓ サポートリクエスト(Web サイト経由のサポート問い合わせフォーム)
www.vector.com/jp/ja/support-downloads/support/

【お問い合わせ時のお願い】

お問い合わせの際、サポート開始前に製品のバージョン(例:CANape バージョン xx.x など)、シリアルナンバーなどをお伺いいたします。お手数ではございますが、事前にご確認のうえ、お問い合わせくださいますようお願い申し上げます。

【サポート対応について】

夜間／休日に頂いたお問い合わせは翌営業日の対応となります。また、内容によりご回答までにお時間を頂く場合がございます。なお、不具合が発生した際にご使用されていた関連プログラム一式のコピーを、検証のためご提供いただくようお願いする場合がございます。何卒ご理解ご協力のほど、よろしくお願い申し上げます。

- ✓ お見積もり／保守契約関連のお問い合わせ

営業部 TEL:(東京) 03-4586-1808 E-mail:sales@jp.vector.com



MORE INFORMATION

Visit our website for:

- > News
- > Products
- > Demo software
- > Support
- > Training and workshops
- > Contact addresses

vector.com