

手作業のテストと検証を自動化する スイッチ&ハードウェアシミュレーション

アンドールシステムサポート(株) / 谷口 正純

1 はじめに

製品のテストと検証を手作業で行っている企業は大変多く、テスト技術者が苦勞している現場を目にすることが多々ある。ソフトウェアの世界では、手作業による人海戦術を自動化する動きが進んでおり、様々なメーカーが自動テストや検証ツールを提供している。そのため、容易にソフトウェアのテストを自動化する環境を構築できる。

しかし、ハードウェアの世界では、依然として手作業による人海戦術が多いのが現状である。特に日本では海外と比べて自動化が遅れている。次のような作業を手作業で行っていないだろうか？

- テスト項目によってケーブルの配線を変更する
- スイッチボックスを使って外部の条件を設定する
- 測定する箇所を変更するためプロービングする

開発現場や製造現場で、写真1のようなスイッチボックスを使った作業を目にすることがある。これこそが典型的な手動テストの例である。テスト項目ごとの手順書を見ながら、スイッチを制御して、テスト結果を記録していることが分かる。テストを自動化する第一歩は、このスイッチボックスによるテストを自動化することである。これらの「手作業」を「自動化」することができたら、様々な課題が解決できるようになる。

今回はテスト現場における、手作業を自動化するための方法を紹介する。

2 手作業によるテストの課題

製品の品質を向上するためにはテストが重要であるが、製品の高機能化に伴い、テスト条件の複雑化が進んでいる。そのため、様々な条件でテストを行うために、テスト項目が増えており、手作業によるテストでは膨大な時間がかかってしまう。

また、作業者の人手不足、熟練度や体調によるテスト品質のバラツキは、製品の品質のバラツキに直結してしまうことになり、不良品を市場に流出してしまうリスクが高まる。主な手作業によるテストのリスクについては、下記のことが考えられる。

- テスト項目が増え、生産性が悪化
- テスト内容により配線の切り替えによる時間のロス
- 作業者の見落とし、勘違いによるボカミス
- 作業者の熟練度や体調によるバラツキ
- テスト結果の入力ミス
- 人手不足によるパフォーマンス低下

驚くべきことに、写真2のような手作業による配線の切り替え、スイッチのON/OFFによる手動のテスト、手作業による測定に依存している企業は多い。その理由は、予算、変更への抵抗、テスト時間を重視していないなどが考えられる。自動化による大きな違いは、配線の切り替えやスイッチのON/OFFの手順をプログラムできることであり、手作業と比較して、より速く、よりミスがなく、より再現性が高いことである。スイッチによるテストの自動化は、予算、規模、将来の拡張性を考慮し、プロジェクト全体の目標に合わせて小規模なものから手軽に導入できる。

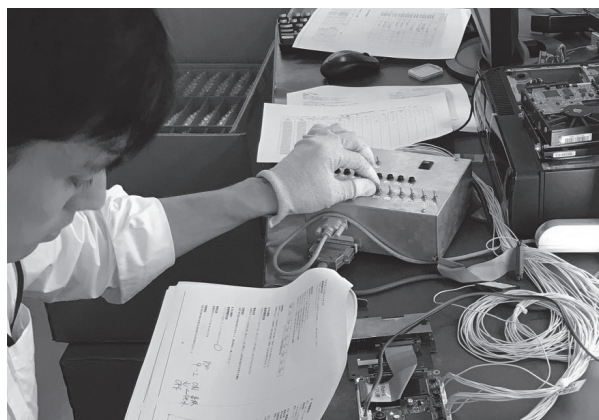


写真1 スwitchボックスを使用したテスト

企業の競争力を向上するためには、限られた人員の負荷を低減して、作業員しかできないことに注力するために、テスト作業の自動化を進める企業が急速に増えている。

自動テストの要求は、主に産業機器、オートモーティブ、アビオニクス、医療、宇宙、通信機器など、高い信頼性と安全性が求められる機器で求められてきたが、現在ではIoT機器、組み込み機器など、あらゆる製品のテストと検証の場面で自動化が進められている。システム全体が複雑化する中で、品質保証、安全性を保証するためには、様々なセンサ、不安定な電源、ケーブルの断線など、特殊な条件下で製品、システムのテストをすることも重要となる。この複雑な条件をシミュレーションモジュールやスイッチングモジュールを利用する事により、効率的に自動テストの環境を構築できる。

自動テストによる製品のテストと検証を実現できると、下記3つのポイントが改善できる。

- 機器の繋ぎ変えを自動化でき、テスト時間を短縮できる
- 同じテストを正確に繰り返し実行でき、テストの正確な再現ができる
- 実環境では作れない過酷な条件や発生頻度が低い条件下でテストできる

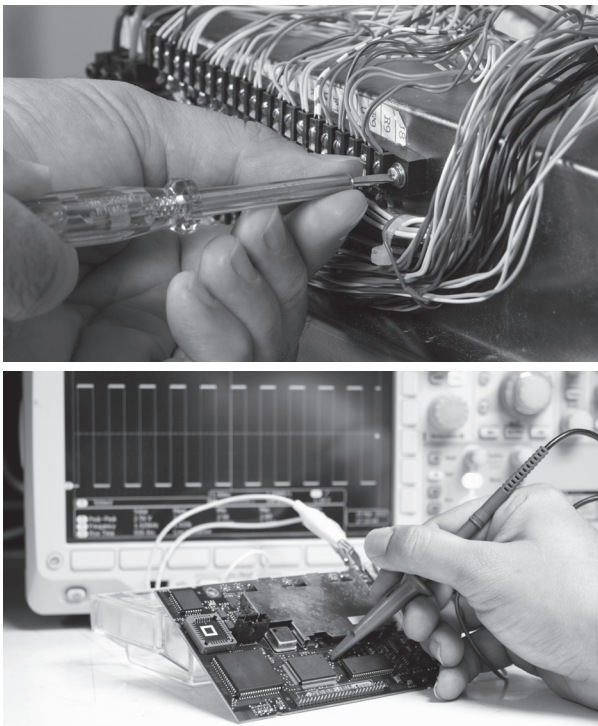


写真2 手作業によるテスト

3 テストを自動化する PXI規格とLXI規格

一般的にテスト作業のすべてを自動化するには、膨大な費用がかかってしまう。実は作業者の「単純作業」と「繰り返し作業」のみを自動化するだけでも、十分な効果を得ることができる。自動化するための仕組みには、GPIB (General Purpose Interface Bus) という計測器の汎用インタフェースがある。GPIBは、1975年にIEEEで国際標準規格化された。しかし、計測器の進化に伴い、通信速度が不十分になったため、今日ではGPIBから、写真3のように、PXI、LXIという新たなオープン規格が世界の主流となり、通信速度の高速化が進んでいる。また、USB、PCIといったパソコンベースのテスト環境もある。

PXI (PCI eXtensions for Instrumentation) は、2005年にPXISA (PXIシステムアライアンス) が規格化した自動テスト用のPCベースのプラットフォームである。高速データインタフェースを利用するためのモジュール式計測システムで、オープン規格を採用しているため、あらゆるメーカーのPXIモジュールとPXIシャーシを混在して使用できる仕様となっている。PXI規格では、機械的、電気的、およびソフトウェアのインタフェースを規定しているため、インテグレーションおよびソフトウェア開発にかかるコストを最低限に抑え、複数メーカーを組み合わせてもトラブルのないテストシステムを構築することができる。

LXI (LAN-based eXtensions for Instrumentation) は、2005年にLXIコンソーシアムにより規格化され、LANを

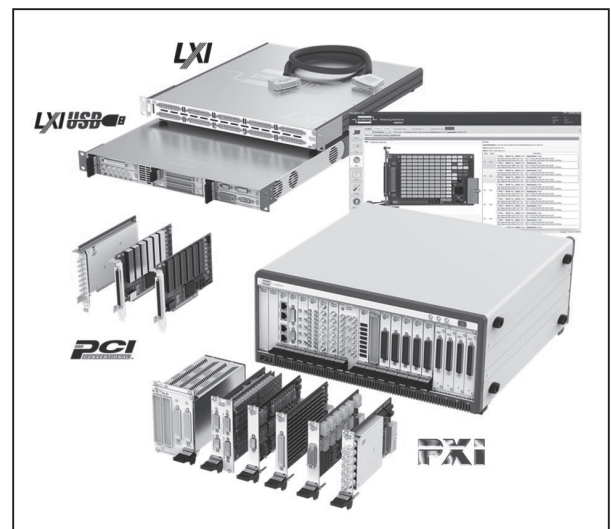


写真3 自動テスト用の標準インタフェース

F E A T U R E

ベースとした自動テストに適した通信規格である。様々なメーカーが開発したLXI準拠の機器を、通信やセットアップの問題なしに連携することを保証している。また、LXI規格は、GPIBやPXIシステムなどの他のテストおよび測定制御システムを補完することを保証している。

4 PXIとLXIの構成

1. PXI、LXIモジュール

自動テストで必要となる配線切り替え用の「スイッチモ

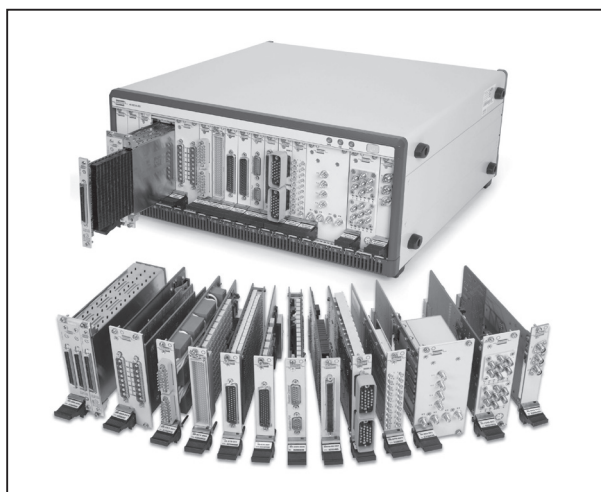


写真4 モジュールベースの自動テスト

ジュール」と、センサ、電源、デジタルI/Oなどをシミュレーションするための「ハードウェアシミュレーションモジュール」は、機能ごとにすべてモジュール化されており、現在は1,800種類を超えている。写真4に示すように、自動化に必要なモジュールをPXIもしくはLXIシャーシに挿入して、自動テスト用のハードウェア環境を構築できる。モジュールの詳細は、<https://www.andor.jp/pickering/download/>より資料をダウンロードできるので参照していただきたい。

従来は製品ごとにテスト環境を自作していた企業にとっては、PXIとLXIモジュールを購入するだけでテスト環境を準備できるため、テスト準備の工数と準備期間を大幅に削減することができる。また、汎用品であるため、メンテナンス性と拡張性に優れており、多品種の共通テスト装置として活用することができる。



写真5 PXIシャーシとLXIシャーシの一例

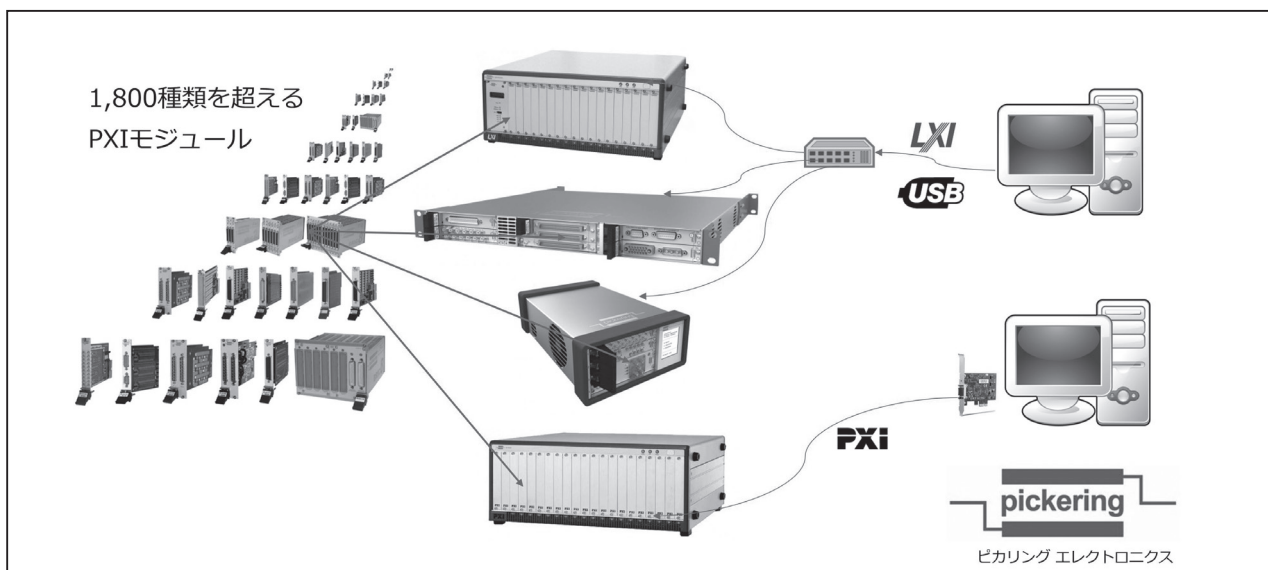


図1 自動テストの標準インターフェース

2. PXI、LXIシャーシ

自動テストに必要なモジュールの枚数に合わせて、シャーシのスロット数を選択する。小規模なシステムであれば2スロットのLXI、USBシャーシが用意されている。自動化するテストシステムの規模に合わせて、写真5のように2スロット、4スロット、6スロット、7スロット、17スロットのシャーシまで用意されている。大規模なテストシステムを構築する場合には、複数台の17スロットシャーシをラックに組み込んで使用することもできる。

3. インタフェース

パソコンから制御するためのインタフェースは、図1のようにPXI、LXIもしくはUSBが標準となっている。PXIの特徴は、PXIコントローラを使ってパソコンからPXIモジュールをコントロールする仕組みであり、ネットワークの影響を受けることなく、1つの筐体のみで大規模な自動テストシステムを容易に構築できる。

LXIの特徴は、パソコンからLANを経由してLXI機器をコントロールする仕組みである。LXIシャーシにはPXIモジュールを組み込んで使用できるため、PXIと同等の拡張性を備えている。ネットワーク上でLXI機器間を接続できるため、事実上の距離に制限がなくなることが大きなメリットとなる。大きな筐体をテストする場合には、大変有効なテストシステムを構築できる。

また、USBによるメリットは、パソコンからUSBを利用してLXIモジュールをコントロールできることである。パソコンと機器間の距離には制限があるが、USBハブを利用して簡単に機器を増設できる。

また、すべてのPXIモジュールとLXIモジュールには、ソフトウェアから簡単に制御できるように、Windows、LabWindows / CVI および LabView、Python用のソフトウェアドライバが付属している。そのため、テストシナリオに合わせた自動テスト用のプログラムは、一般的なソフトウェア開発環境を利用したことがあるソフトウェアエンジニアであれば、誰でも容易に開発することができる。

5 自動テストの構成例

自動テストの基本的な構成を図2に示す。テスト対象の基板を中心として、スイッチモジュールとハードウェアシミュレーションモジュールが接続されていることが分かる。測定すべきテストポイントはマトリックススイッチと接続されており、スイッチのON/OFFによってテスト項目に合わせて測定ポイントと計測モジュールの経路が自動的に切り替わる。測定器もPXIモジュールやLXI対応のものを選択することにより、測定器の設定から測定結果の記録まで自動化することができる。

充電機を利用する機器のテスト条件には、充電機の満充電、放電中、完全に放電している状態、バッテリーの異常状態のときを考慮したテストを行う必要がある。実物の電池を使わずに、充電機のあらゆる状態を瞬時に再現するために、バッテリーシミュレーションを利用することができる。実物の電池を使わずに、電池の充電状態をソフトウェアから設定することができる。

次にセンサを利用する機器のテストであるが、温度センサであれば、大きな環境試験装置がなければ高温から低温まで正確に変化させることは難しい。しかし、センサシミュレーションを利用すると、ソフトウェア制御により瞬時に指定した温度のセンサをシミュレーションさせることができる。センサシミュレーションは、プログラム抵抗で構成されており、温度センサ、ひず

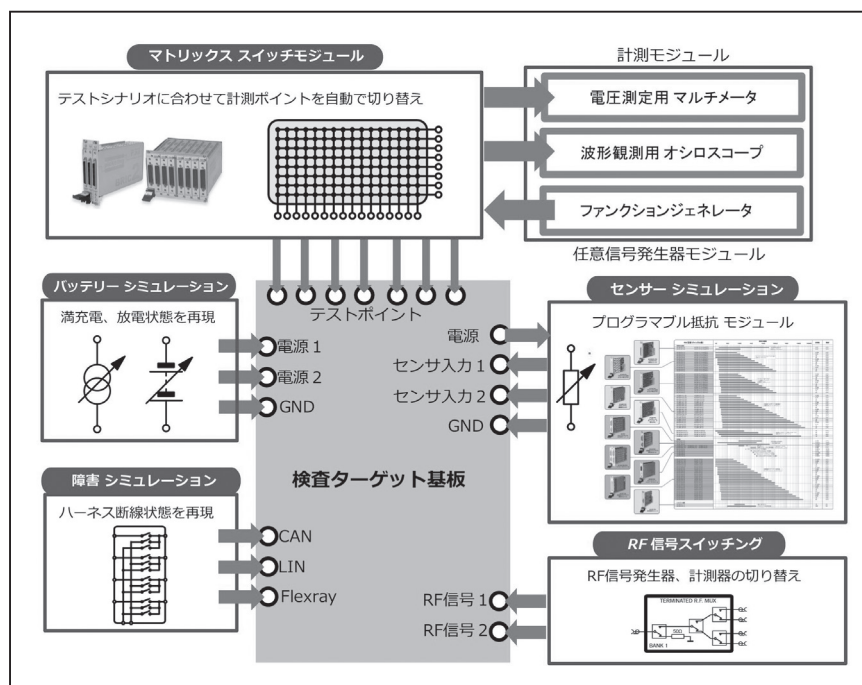


図2 自動テストの構成例

F E A T U R E

みゲージセンサ、熱電対センサなど、様々なセンサを模擬することができる。センサの代わりに、ボリューム抵抗を複数使用してセンサのシミュレーションをさせている企業が多い。しかし、ボリューム抵抗は手作業で抵抗値を調整しなければならず、1つの条件を作るために時間がかかってしまう。そのため、複数のセンサをシミュレーションする用途には向いていない。

マイクロウェーブを利用する機器のテストには、RF信号用のスイッチが用意されている。様々な条件のRF信号発生器や計測器を自由に切り替えながら、テストを行うことができる。また、機器の障害状態をシミュレーションするための障害シミュレーションが用意されている。この障害シミュレーションは、テスト対象の機器のケーブルが断線、電源とのショート(天絡)、GNDとのショート(地絡)、信号間のショートした状態をシミュレーションするものである。ハードウェアの異常状態に機器の安全性をテストすることができ、人命に関わるあらゆる製品の安全性テストに活用することができる。

このように、テスト対象の機器の周辺環境をシミュレーションすることにより、テスト条件を自動的に変化させ、測定すべきテストポイントと計測器の経路をマトリックススイッチで切り替えながら自動テストすることができる。ハードウェアのテストだけではなく、ソフトウェアの検証用としても多く使われており、ソフトウェアを修正するたびに、すべてのテスト項目をやり直す場合に、自動テストシステムを構築して1クリックでテストを完了できるメリットは非常に大きいのではないだろうか。

6 センサ搭載機器のテストを自動化する方法

あらゆる電子機器には、制御している機器を監視するために、センサを利用している。センサには多種多様なものがあり、温度センサ、ポテンショメータ、歪みゲージセンサ、圧力センサ、加速度計、距離センサ、変異センサ、フォトランジスタ、フォトダイオードなどがある。テストをする場合には、センサ値の上限値、下限値などを作って、テスト対象を動作させる必要がある。実際のセンサからハードウェアシミュレーションに置き換え、実際のセンサと同じ動作を正確にシミュレートさせることができれば、ソフトウェアから自由にセンサの値を設定して、実環境では難しい条件を容易に設定することができる。

センサのシミュレーションには、図3のようなプログラム抵抗モジュールを使うことができる。

プログラム抵抗モジュールには、短絡から数百万オームまでの幅広い抵抗範囲のモジュールが用意されている。抵抗設定の分解能は、最小0.03%の精度で最小2mΩの分解能となっており、広範囲のアプリケーションで利用できるよう10～22MΩまで選択できる。

センサは、大きく分けて3つの特性に分けられる。主に「抵抗値が変化するセンサ」としては、ポテンショメータ、ひずみゲージ、温度計がある。「電圧が変化するセンサ」は、変異センサ、圧力センサ、加速度計があり、「電流値が変化するセンサ」は、フォトランジスタ、フォトダイオードなどがある。

図4のように、センサシミュレーションを使用することにより、テストとデバッグ時にセンサの値をソフトウェアから自在に制御することができ、テスト対象の機器の動作をテストすることができる。ソフトウェアにより、複雑な条件を設定してテストできるため、テスト時間の削減とテスト品質の改善が期待できる。

7 ハードウェア障害テストを自動化する方法

安全性が求められる電子機器は、ハードウェアの障害状態を再現したテストが必要である。システム障害のテストは新しいものではなく、製品を検証する際の重要な側面であり、システムに電気的な障害を挿入するテストを行う。テストプロセスでは、通常状態、腐食状態、電源と短絡、GNDと短絡、オープン(断線)回路、およびその他の電氣的故障が経年、損傷、または設置不良の影響によって生じる可能性のある様々な条件を再現する必要がある。しかし、多くの企業では、実際にケー

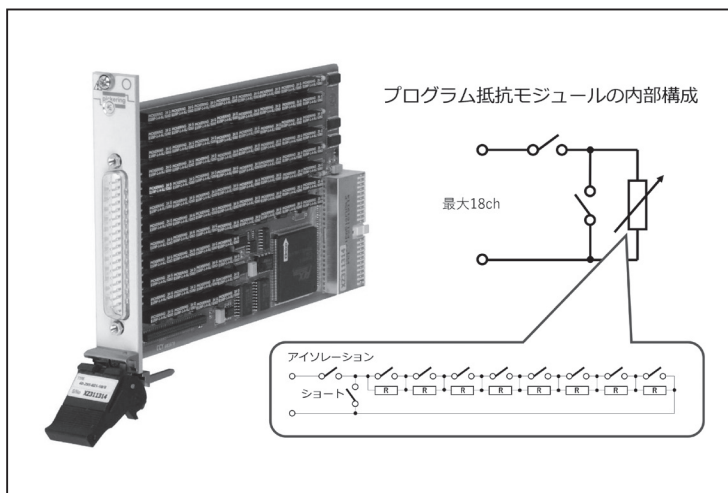


図3 プログラム抵抗

ブルを断線させたり、短絡させたりして、実機を使って手作業でテストを行っている。このケーブルの障害状態をシミュレーションして、自動化したものが、図5の障害挿入モジュール(フォルトインサージョンモジュール)である。

障害挿入モジュールの内部は、図6のように複数のスイッチから構成されている。センサや通信ユニットと被検査基板の間に、障害挿入モジュールを中継するように接続する。U1の信号は障害挿入モジュールのスイッチがONにされており、ケーブルで直接接続されている状態と等しく、障害がない状態に設定されている。次にU2の信号は、障害挿入モジュールのスイッチがOFFされているため、ケーブルが断線した状態と等しくなり、断線状態のテストを行うことができる。

また、U3の信号は障害挿入モジュールのスイッチがONされているが、障害バスのスイッチもONされており、障害バスのF4と接続されていることが分かる。F4にはGNDが接続されているため、U3の信号とGNDがショートするように設定されていることが分かる。この状態は、U3を地絡した場合のテストを行うことができる。このように、実際にケーブル、ハーネスを切断せずに、ケーブルの不具合をソフトウェアからスイッチのON/OFFにより制御することができるため、ハードウ

エアの障害発生時の回路検証、ソフトウェア検証に必要なテストの条件を瞬時に設定することができるようになる。

8 まとめ

今回で紹介した内容が手作業によるテストを自動化するためには、「業務で行っている繰り返しの単純作業を自動化できないか?」という視点をもつことが重要である。最初からすべての作業を自動化しようと考え、プロジェクトの規模が拡大し、高額な予算が必要になってしまう。しかし、小さな作業に目を向けて、作業員の手が離せないことだけを自動化しようと思えば、小規模のLXIシステムを活用して容易に作業を自動化することができる。

スイッチによる自動化のアイデアは、小冊子「ピカッとわかる!スイッチ活用アイデア集1」にまとめたものをWebでも公開している(<https://www.andor.jp/pickering/download/>)。

今回の記事と合わせて、ご参考にして頂けると幸いです。

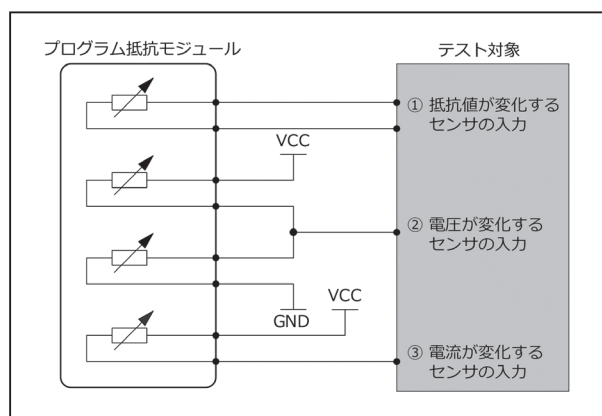


図4 センサシミュレーションの使い方

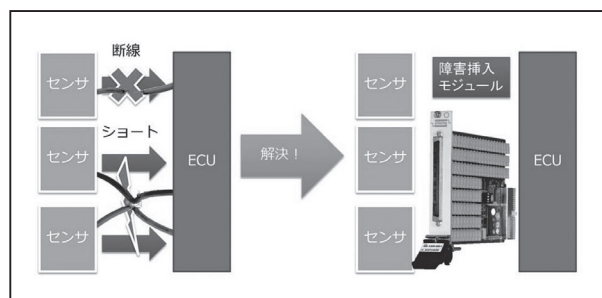


図5 障害挿入モジュール

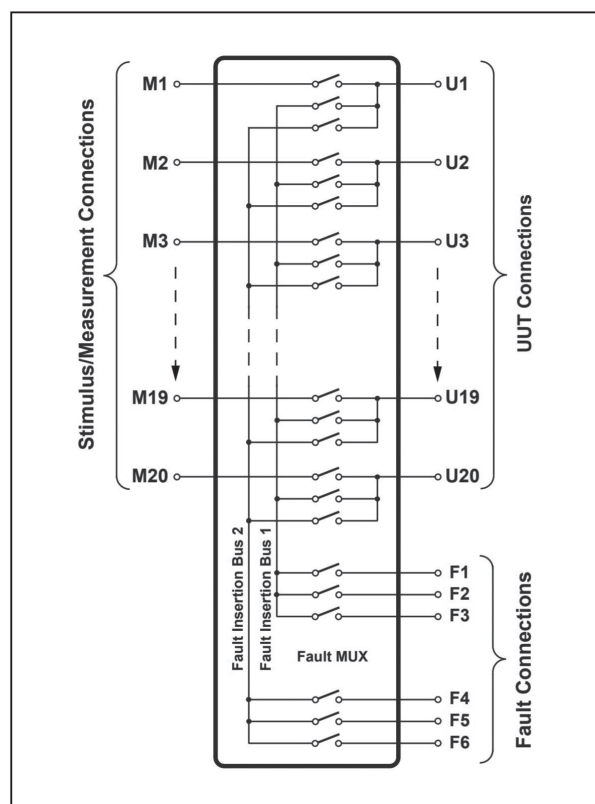


図6 障害挿入モジュールの内部構成