

Frei von allen Fehlern



Fehler in komplexen Testfällen zu finden, gleicht der Suche nach der berühmten Nadel im Heuhaufen. Zur Vereinfachung der Fehlersuche bietet AutomationDesk 2.2 etwas Neues: den Debugger. Mit ihm können die Testabläufe Schritt für Schritt ausgeführt und überprüft werden. Fehlerquellen lassen sich so gezielt finden und eliminieren.

In der Hardware-in-the-Loop (HIL)-Simulation hat die Testautomatisierung ihren festen Platz. Die steigende Komplexität der Steuergeräte-Funktionen führt jedoch auch unmittelbar zu komplexeren Testfällen.

AutomationDesk-Version 2.2, die Testautomatisierungssoftware von dSPACE, erleichtert die Überprüfung dieser umfangreichen Testfälle. Die Funktion „Find Inconsistencies“ führt bereits vor der Testausführung statische Analysen durch. Mit dem neuen Debugger lassen sich die Testabläufe bereits während der Implementierung Schritt für Schritt prüfen und korrigieren.

Den Test testen

In AutomationDesk werden Testabläufe grafisch beschrieben und bestehen aus einer Vielzahl einzelner Testschritte, den sogenannten Blöcken. Bereits vor der Testausführung kann der Anwender mit „Find Inconsistencies“ Fehler fin-

Der Debugger ermöglicht eine schnelle und effiziente Fehlersuche, da potenzielle Fehlerquellen gezielt betrachtet werden können.

den und korrigieren. Dafür wird die Teststruktur mitsamt den definierten Datenobjekten, Referenzen und Code-Fragmenten analysiert, sodass beispielsweise offene Referenzen oder Syntaxfehler automatisch erkannt werden.

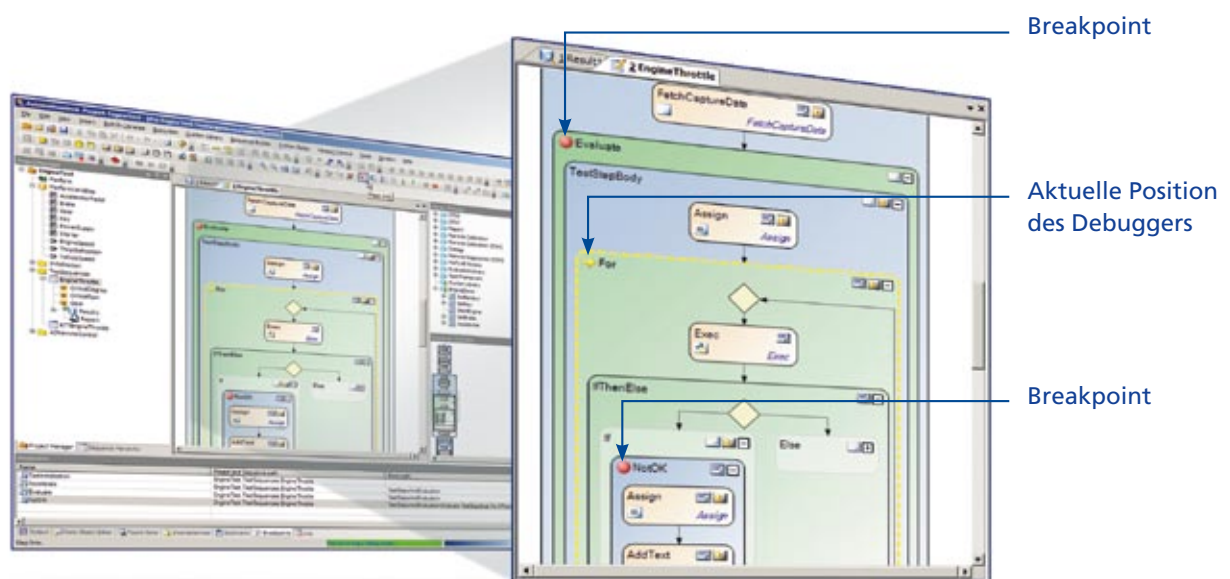
Im nächsten Schritt wird der Test im Offline-Modus ausgeführt, um Fehler im Testablauf zu finden, ohne dass die komplette Testumgebung, z. B. HIL-Simulator oder Kalibrierwerkzeuge, vorhanden sein muss. Hierbei werden Default-Werte definiert, die bei Zugriffen auf nicht vorhandene Hard- oder Software zurückgeliefert werden. Typische Fehler, die sich hier finden lassen, sind Fehler in der Ablauf- oder Kontrollstruktur, wie z. B. Fehler in den Bedingungen von If-Else-Ver-

zweigungen oder in Schleifen, falsche Parametrierungen oder Laufzeitfehler. Wurden alle diese Punkte korrigiert, folgt die Testausführung am HIL-Simulator. Durch den Offline-Modus steht der HIL-Simulator während der Testentwicklung für die Ausführung anderer Tests zur Verfügung.

Debugging am PC

Im Fall von unerwarteten Testergebnissen oder Fehlern in der Testausführung – offline am PC oder auch online am HIL-Simulator – hilft der Debugger, die Ursache durch die schrittweise Ausführung gezielt ausfindig zu machen.

Mit dem Debugger kann der Entwickler den Testablauf an beliebigen Stellen mit sogenannten Breakpoints



An jedem Breakpoint stoppt der Debugger automatisch die Testdurchführung.



Mit der Debugger-Symboleiste kann der Anwender den Testablauf Schritt für Schritt prüfen.

markieren. Startet er die Sequenz im Debug-Modus, stoppt die Ausführung automatisch an den markierten Stellen. Folgende Möglichkeiten stehen dann in der Debugger-Symboleiste zur Verfügung:

- Testsequenz bis zum nächsten Breakpoint weiterlaufen lassen
- Blöcke schrittweise ausführen
- Bei komplex strukturierten, hierarchischen Testblöcken kann der Entwickler auch eine Schicht tiefer einsteigen und diese ausführen oder auf der aktuellen Hierarchieebene bleiben und sie als Ganzes ausführen.

Die aktuellen Variablenwerte sind während des Debuggings jederzeit einsehbar und können bei Abweichungen verändert werden, um den weiteren Ablauf mit korrekten Werten zu testen. Schritt für Schritt kann der Anwender den Test so mit dem Debugger prüfen und bei Bedarf korrigieren.

Effiziente Prozesse

Nach Abschluss des Debugging-Vorgangs kehrt der Anwender in den normalen Ausführmodus zurück und startet seine Testsequenz erneut. Die Schritt-für-Schritt-Ausführung ermöglicht eine schnelle

und effiziente Fehlersuche, da potenzielle Fehlerquellen gezielt betrachtet werden können. Der geschickte Einsatz von „Find Inconsistencies“, dem Offline-Modus und dem Debugger beschleunigt die Testentwicklung, erhöht die Testqualität und verbessert insgesamt den Testprozess. ■