

Configuration Desk

Neue Flexibilität
in den Arbeitsabläufen

ConfigurationDesk ist die Konfigurationssoftware für den neuen Simulator SCALEXIO. ConfigurationDesk und SCALEXIO setzen neue Maßstäbe für Flexibilität und Wiederverwendung bei der Hardware-in-the-Loop-Simulation.

The screenshot shows the ConfigurationDesk interface. On the left, a block diagram of an external device is visible with sections for ESP, COMM, DI, and POWER. On the right, a 'Table Window 1' displays a hierarchy of components. Below the table, a signal trace for 'BFFS_SIG[V]' is shown.

Name	Description	Pin(s)	Port Type	Physical Attribute	Reference Port(s)	Short to GND	Short b...
HeatSink							
ESP							
COMM							
M_ESP	ESP/ASR button	A13	In	Voltage	POWER(KL30V)	Not allowed	Not allo
BRL1	Brake Light Switch	A41	In	Voltage	POWER(GND_V)	Not allowed	Not allo
BFFS_NC	Release Switch NC (normally closed)	A30	In	Voltage	POWER(BFFS_SIG)	Not allowed	Not allo
BFFS_NO	Release Switch NO (normally open)	A27	In	Voltage	POWER(BFFS_SIG)	Not allowed	Not allo
HES	Handbrake Switch	A39	In	Voltage	POWER(GND_V)	Not allowed	Not allo
POWER							
BFFS_SIG	BFFS (BFA Booster) supply	A28	Out	Voltage	D(BFFS_NC); D(BFF...	Allowed	Allowec
BST_PWR	Highside for Booster (supply)	A17	Out	Voltage	POWER(GND_V)	Not allowed	Not allo
CLUS_SP	Sensor Cluster supply	A06	Out	Voltage	POWER(CLUS_GND)	Allowed	Allowec
CLUS_GND		A24	Reference	Voltage	POWER(CLUS_SP)	Not allowed	Not allo
KL30P	KL30	A01	In / out	Undefined	POWER(GND_P)	Not allowed	Not allo
MP_Sensp	Membrane Sensor supply	A26	Out	Voltage	POWER(MP_GND)	Not allowed	Not allo
MP_GND		A05	Reference	Voltage	AT(MPL_SIG; POWE...	Not allowed	Not allo
KL30V	KL30	A32	In / out	Undefined	D(MPL_ESP)	Not allowed	Not allo
IGN	KL15	A04	In / out	Undefined	D(BRL_OUT)	Not allowed	Not allo
GND_V	KL31 for A32	A16	In	Undefined	D(BRL1); D(HES; P...	Not allowed	Not allo
		A19	Out	Voltage	POWER(BFFS_SIG)	Allowed	Allowec

Abbildung 1: Interface-Beschreibung der angeschlossenen Geräte.

Veränderte Anforderungen

Im Bereich der Hardware-in-the-Loop (HIL)-Simulation sind mittlerweile in fast allen Unternehmen spezialisierte Abteilungen entstanden, die jeweils Teile von Projekten (Modelle, I/O-Konfiguration etc.) entwickeln und pflegen. Diese Teilkomponenten müssen flexibel zu einem Gesamtprojekt zusammengefügt und das so entstandene HIL-System muss vollständig dokumentiert werden. Die Dokumentation beinhaltet die angeschlossenen Geräte (Elektronische Steuerungen (ECUs) und Echtlasten), den Kabelbaum zwischen HIL-System und ECUs, die verwendeten HIL-Hardware-Komponenten mit deren Konfiguration und das zum Test verwendete Umgebungsmodell.

Projektverwaltung mit ConfigurationDesk

Mit ConfigurationDesk® hat dSPACE eine Software entwickelt, mit der die oben beschriebenen Anforderungen in einer intuitiven grafischen Oberfläche erfüllt werden. Die jetzt neu verfügbare Version (Implementation Version) ist der Nachfolger des bekannten Real-Time Interfaces (RTI) zur Hardware-Konfiguration und zur Implementierung der Echtzeitappli-

kationen auf der neuen HIL-Technologie von dSPACE SCALEXIO®. Mit ConfigurationDesk wird die SCALEXIO-Hardware vollständig konfiguriert, d.h., Einstellungen per Hardware-Jumper o.ä. sind nicht erforderlich. Darüber hinaus dokumentiert ConfigurationDesk das gesamte HIL-System, bestehend aus externen Geräten, Kabelbaum, verwendeter dSPACE Hardware mit deren Konfiguration und dem Umgebungsmodell samt Schnittstellen zur I/O.

Interface-Beschreibung der angeschlossenen Geräte

Am Anfang eines HIL-Projekts steht die Definition des elektrischen Interfaces der zu testenden ECU. Hier sind Informationen über Stecker- und Pin-Anzahl relevant. Aber auch abstraktere Daten werden benötigt, wie logische Signalnamen mit Beschreibung, Signalrichtung (Eingang oder Ausgang der ECU) und physikalische Signaleigenschaften (Spannungs- oder Stromsignal) sowie Informationen dazu, ob auf einem ECU-Pin ein bestimmter elektrischer Fehler geschaltet werden darf und welche Lasten die ECU an diesem Pin erwartet. Diese Daten können direkt im Tool eingegeben (Abbildung 1)

oder über eine Microsoft®Excel®-Liste importiert werden.

Vorbereitung der Umgebungsmodelle

Die Umgebungsmodelle stehen meist als MATLAB®/Simulink®-Modelle zur Verfügung. Für die Ankopplung der I/O an das Modell stellt dSPACE eine Simulink-Bibliothek zur Verfügung, die so genannte Model-Port-Blöcke für Eingangs-, Ausgangs- und Trigger-Signale enthält (Abbildung 2). Diese Blöcke können im Modell auf allen Modellebenen eingefügt werden. Liegen alle Signale zur I/O-Ankopplung auf oberster Ebene, können alternativ auch Standard-Simulink-In- bzw. -Out-Blöcke verwendet werden. An diesen Interface-Blöcken endet zunächst die Signalkette. Im Simulink-Modell sind also keinerlei hardware-spezifische Informationen abgelegt. Dies ermöglicht eine sehr einfache Wiederverwendung desselben Modells in unterschiedlichen HIL-Projekten.

Von der ECU zum Modell

In ConfigurationDesk werden alle Komponenten zu einem Gesamt-HIL-Projekt zusammengefügt. In einer übersichtlichen Drei-Spalten-Ansicht

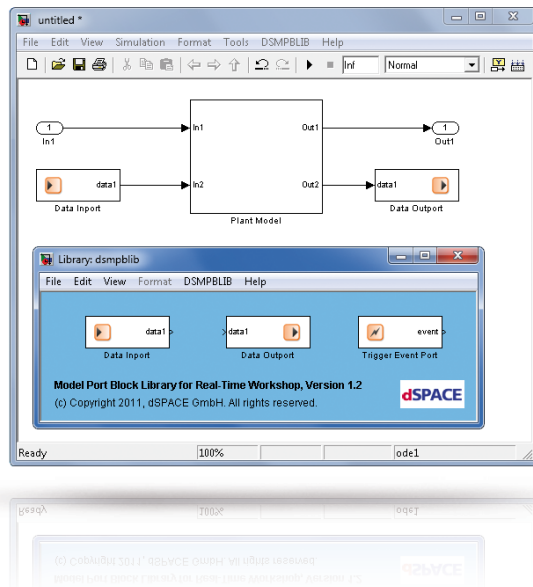


Abbildung 2: Simulink-Modell-Interface.

werden die Interfaces der angeschlossenen Geräte (linke Spalte) mit I/O-Funktionalitäten des HILs (mittlere Spalte) und diese wiederum mit Modellschnittstellen (rechte Spalte) verknüpft. Das Tool ist in der Lage, zu den externen Geräten passende I/O-Funktionalitäten vorzuschlagen, z.B. empfiehlt es für einen ECU-Pin, an dem ein von der ECU generiertes Spannungssignal anliegt, eine HIL-I/O-Funktion zur Spannungsmessung. Alternativ können I/O-Funktionen auch aus einer Bibliothek heraus ausgewählt werden. Das Werkzeug weist den Benutzer auf eventuelle Fehlverknüpfungen hin, wenn beispielsweise ECU-Ausgänge versehentlich mit HIL-Ausgängen statt mit -Eingängen verknüpft wurden. Die I/O-Funktionen werden in einem weiteren Arbeitsschritt mit Modellschnittstellen verbunden (rechte Spalte). Hier gibt es zwei alternative Vorgehensweisen: ConfigurationDesk kann entweder eine zu den I/O-Funktionen passende Modell-Interface-Beschreibung generieren und in ein Simulink-Modell exportieren, oder der Anwender bereitet das

Modell in Simulink durch Einfügen von Model-Port-Blöcken vor. Durch eine Modellanalyse wird die Interface-Beschreibung dann in ConfigurationDesk übernommen. Die I/O-Funktionen definieren zunächst nur, welche Funktionalität vom Simulator durchgeführt werden soll (z.B. PWM-Signal-Generierung), unabhängig davon, auf welcher konkreten SCALEXIO-Hardware diese Funktionalität zur Verfügung gestellt wird. Die gleiche Funktionalität kann in den meisten Fällen auf unterschiedlichen Hardware-Kanälen bzw. Kanaltypen bereitgestellt werden. In einem weiteren Konfigurationsschritt, dem so genannten Hardware Resource Assignment, werden den I/O-Funktionen daher die benötigten Hardware-Kanäle zugewiesen. Dies kann manuell oder automatisch durch ConfigurationDesk erfolgen. Wenn Signaleigenschaften, z.B. die maximale Stromaufnahme, nicht mit einem einzelnen Hardware-Kanal abgedeckt werden können, bestimmt ConfigurationDesk die benötigte Kanalanzahl des gewünschten Kanaltyps und ordnet diese der I/O-Funktion zu.

Berechnung des Kabelbaums

ConfigurationDesk berechnet optional nicht nur die Verbindungsinformationen zwischen HIL-System und ECU, sondern auch zwischen dem HIL-System und anderen extern angeschlossenen Geräten wie den Echtlasten. Hierbei werden in einer Excel-Liste alle zu verbindenden Pins aufgeführt, auch Verbindungen zwischen Simulator-Pins oder direkte Verbindungen zwischen ECUs und Echtlasten, wodurch der Aufbau des realen Kabelbaums sehr einfach wird.

Steuerung des Build-Prozesses

Alle Phasen des Build-Prozesses werden von ConfigurationDesk aus gesteuert. Der Modell-C-Code wird durch MATLAB/Simulink erzeugt. Für die I/O-Funktionalität generiert ConfigurationDesk eigene Code-Anteile. Es können auch einzelne Phasen gezielt abgeschaltet werden. Wenn zum Beispiel nur die I/O-Konfiguration verändert wurde, kann Generierungszeit eingespart werden, indem die Modell-Code-Generierung deaktiviert und „alter“ Modell-Code wiederverwendet wird. Nach der C-Code-Generierung werden alle Code-Anteile kompiliert und miteinander verlinkt. Daraus wird eine Echtzeitapplikation erzeugt, die auf ein SCALEXIO-System geladen und dort ausgeführt werden kann.

Flexibilität der Arbeitsschritte

Die beschriebene Reihenfolge der Arbeitsschritte ist nur eine von vielen möglichen. Auch sind nicht alle Elemente der Signalkette zwingend erforderlich. So sind Informationen über die angeschlossenen Geräte für die eigentliche HIL-Simulation nicht notwendig. Ein Build-Prozess kann auch durchgeführt werden, wenn nur I/O-Funktionen und Verhaltensmodell vorhanden sind. Für die Berechnung des externen Kabelbaums und zu Dokumentationszwecken wird die linke Spalte der Drei-Spalten-Ansicht benötigt.

Übersicht im Projekt

Bei HIL-Projekten kommen schnell große Mengen an Signalketten zusammen. Hier kann eine Konzentration auf einzelne Teilbereiche sinnvoll sein. Hierfür wurde in ConfigurationDesk die Möglichkeit zur Sichtbildung geschaffen. Mit wenigen Mausklicks kann man sich beispielsweise nur die Signalketten anzeigen lassen, die eine Spannungsvermessung beinhalten, oder eine Sicht mit Signalketten erstellen, die mit einem bestimmten ECU-Stecker verbunden sind. Natürlich existiert auch immer eine globale Sicht, die alle Elemente des Projekts enthält.

Wiederverwendung von Teilkomponenten

Die Teilkomponenten eines Projekts sind sehr einfach wiederverwendbar (Abbildung 3). So kann zum Beispiel

eine ECU-Beschreibung in ein neues Projekt importiert und dort angepasst werden. Auch das gleiche Simulink-Modell lässt sich problemlos für verschiedene Projekte verwenden. Da hier keinerlei Informationen über die I/O-Hardware gespeichert sind, können auch ganz andere I/O-Funktionalitäten mit den gleichen Modellschnittstellen verbunden werden. Alle Teilkomponenten werden in separaten XML-Dateien gespeichert und lassen sich so mit einem Versionsmanagement-Tool verwalten. ■

Die Vorteile auf einen Blick

- Mit ConfigurationDesk ist es sehr einfach möglich, in verteilten Teams zu arbeiten, in denen beispielsweise die Simulink-Modellierung getrennt und unabhängig von der Hardware-Konfiguration erfolgt.
- Der modulare Aufbau der ConfigurationDesk-Projekte ermöglicht eine einfache Wiederverwendung von Teilkomponenten.
- Die flexible SCALEXIO-Hardware kann durch die getrennte Konfiguration der Funktionalität und der hierfür verwendeten Hardware optimal ausgenutzt werden.
- Das gesamte HIL-Projekt wird in ConfigurationDesk vollständig dokumentiert.

Abbildung 3: Signalkette am HIL und zugeordnete Teilkomponenten in ConfigurationDesk.

