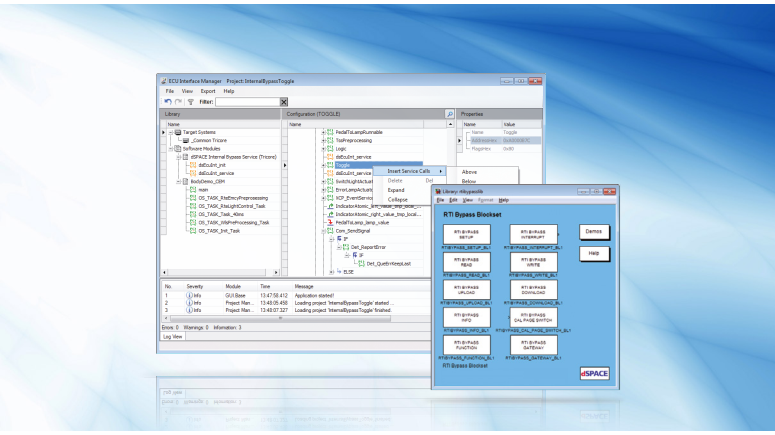
ECU Interface Base Package

External and on-target bypassing

Highlights

- Provides real-time access to production ECUs during RCP, HIL, and field test
- Allows efficient function bypassing directly on existing ECUs or together with separate RCP hardware
- Allows to prepare ECU interfacing on the basis of source code or binary code with the optional Binary Code Management module



No.	Severity	Module	Time	Message
1	Info	GUI.Base	13:47:58.412	Application started!
2	Info	Project Man...	13:48:05.458	Loading project 'InternalBypassToggle' started ...
3	Info	Project Man...	13:48:07.327	Loading project 'InternalBypassToggle' finished.

Errors: 0 Warnings: 0 Information: 3

Application Areas

For rapid control prototyping (RCP), hardware-in-the-loop (HIL), function-in-the-loop (FIL), vehicle-in-the loop (VIL), and ECU (electronic control unit) field tests, real-time access to ECUs is crucial. For a guaranteed real-time access, the existing ECU code has to be prepared to allow additional functionalities to be mapped to it. These functionalities can be executed by using external real-time hardware or even directly on the original ECU (on-target). When developing new ECU software functionalities, an incremental development based on existing ECU hardware and software is common. The dSPACE tool chain for ECU interfacing allows new functionalities to be mapped into the existing ECU by using the external or the on-target bypassing approach. When testing a production ECU with a HIL system, internal states of the ECU can be captured synchronously to become part of the running real-time test. Furthermore, very complex, unavailable, or unaccessible sensors can be simulated. Even partial or complete ECU software tests (FIL in the laboratory and VIL in the vehicle) are possible by directly connecting a plant or environment simulation that runs on real-time hardware to the ECU without the need to connect the full I/O. During a field test, the functionality of an ECU is tested in a real environment. In case an error is detected in the ECU software (e.g., incorrect internal variable value), it is possible to quickly fix it by using on-target bypassing. Afterwards, the test-drive can continue. To cover all these scenarios, dSPACE provides a comprehensive ECU interface software tool chain which is based on the ECU Interface Base Package.

Key Benefits

The dSPACE ECU Interface Software allows for interfacing to existing ECUs without knowledge of the original implementation details. It differentiates between preparing the ECU software for ECU interfacing and implementing new functionality as well as external and on-target bypassing. The ECU Interface Base Package, which contains the ECU Interface Manager for software preparation and the RTI Bypass Blockset for function implementation, enables the access to ECUs that have been already prepared for ECU interfacing with external dSPACE hardware. An optional Binary Code Management Module lets you prepare the binary code of a microcontroller for ECU interfacing. The optional On-Target Module together with the target compiler enables the RTI Bypass Blockset to build an application of a new model-based function for the microcontroller. Therefore, it merges the new function code with existing ECU code to run completely on the target ECU, without extra hardware. For optimization, it allows you to use fragmented memory areas and even free up space by overwriting unused functions. This way, you can use the limited ECU resources more efficiently. To increase efficiency even further or if there is a strong focus on production, you can use the certified production code generator TargetLink.

Functionality Overview

Module	Description
ECU Interface Base Package	- Contains the ECU Interface Manager (p. 3) for software preparation and the RTI Bypass Blockset (p. 3) - Supports external ECU interfacing on the following dSPACE systems: - Non-SCALEXIO systems, i.e., systems for which the real-time application is implemented with RTI and the RTI Bypass Blockset - SCALEXIO systems, i.e., systems, for which the real-time application is implemented with ConfigurationDesk - Supports on-target ECU bypassing, i.e., applications implemented with RTI and the RTI Bypass Blockset that are executed on the target ECU - Import, analysis and examination of binary ECU code - Provides dSPACE ECU Services that support the DCI-GSI2 and generic XCP for ECU Interfacing
Binary Code Management Module (target-specific)	- Optional module for the ECU Interface Base Package - Insertion of dSPACE ECU Services on a binary code basis for external and on-target ECU interfacing - Available for Infineon TriCore (including the latest AURIX 2G/TC3xx), NXP (Freescale) MPC5xxx, STMicroelectronics SPC5xxx, and Renesas V850/RH850
On-Target Module (target-specific)	- Optional module for the ECU Interface Base Package - Implementation of Simulink and TargetLink bypass functions on a specific target ECU with existing dSPACE ECU services - Requires an additional compiler (target-specific) - Available for Infineon TriCore (including the latest AURIX 2G/TC3xx), NXP (Freescale) MPC5xxx, STMicroelectronics SPC5xxx, Renesas V850/RH850, and ARM Cortex R4/R5
Compiler (target-specific)	Depending on the microcontroller, one of the following C compilers is required: - HighTec compiler (target-specific): For Infineon TriCore (including the latest AURIX 2G/TC3xx), NXP (Freescale) MPC5xxx, STMicroelectronics SPC5xxx, and Renesas V850/RH850 - GNU compiler collection: For ARM Cortex R4/R5

Order Information

Product	Order Number
ECU Interface Base Package	- EIF_BASE

Relevant Software and Hardware

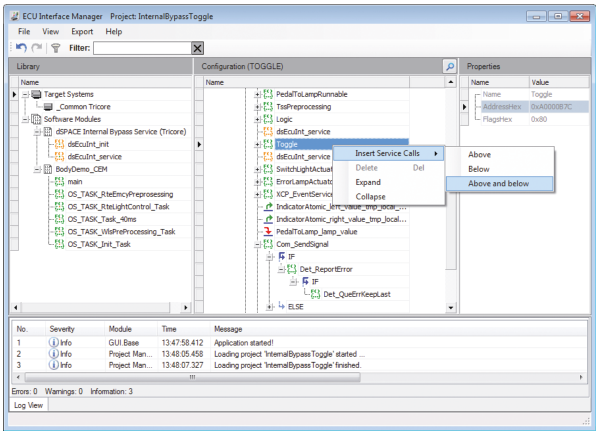
Software		Order Number
Optional	- Binary Code Management Module for Infineon TriCore (including the latest AURIX 2G/TC3xx)	- EIF_BCM_TRICORE
	- Binary Code Management Module for NXP (Freescale) MPC5xxx and STMicroelectronics SPC5xxx	- EIF_BCM_MPC5xxx
	- Binary Code Management Module for Renesas V850/RH850	- EIF_BCM_V850X
	- On-Target Module for Infineon TriCore (including the latest AURIX 2G/TC3xx)	- EIF_OT_TRICORE
	- On-Target Module for NXP (Freescale) MPC5xxx and STMicroelectronics SPC5xxx	- EIF_OT_MPC5xxx
	- On-Target Module for Renesas V850/RH850	- EIF_OT_V850X
	- HighTec GNU CC Compiler for Infineon TriCore (including the latest AURIX 2G/TC3xx)	- HIGHTECC_TRICORE
	- HighTec GNU CC Compiler for NXP (Freescale) MPC5xxx and STMicroelectronics SPC5xxx	- HIGHTECC_MPC5xxx
	- HighTec GNU CC Compiler for Renesas V850/RH850	- HIGHTECC_V850X
	- dSPACE ECU Flash Programming Tool	- DSPACE_ECU_FLASH
	- TargetLink	- See relevant product information

Hardware		Order Number
Optional	- Generic Serial Interface DCI-GSI2	- DCI_GSI2
	- Additional Bypass Interface Solutions	- See relevant product information
	- MicroAutoBox II / MicroAutoBox III	- See relevant product information
	- PHS-based systems	- See relevant product information
	- SCALEXIO systems	- See relevant product information

Components of the ECU Interface Base Package

ECU Interface Manager

The ECU Interface Manager, which is part of the ECU Interface Base Package, allows you to configure the access to your ECU. It also provides you with the ability to apply modifications required for ECU Interfacing to the ECU application's HEX file with the optional Binary Code Management module. The ECU Interface Manager offers an easy-to-understand, list-based view and, for more complex preparation tasks, also a comprehensive structural visualization of the ECU application (e.g., function calls, variable accesses, conditional areas).



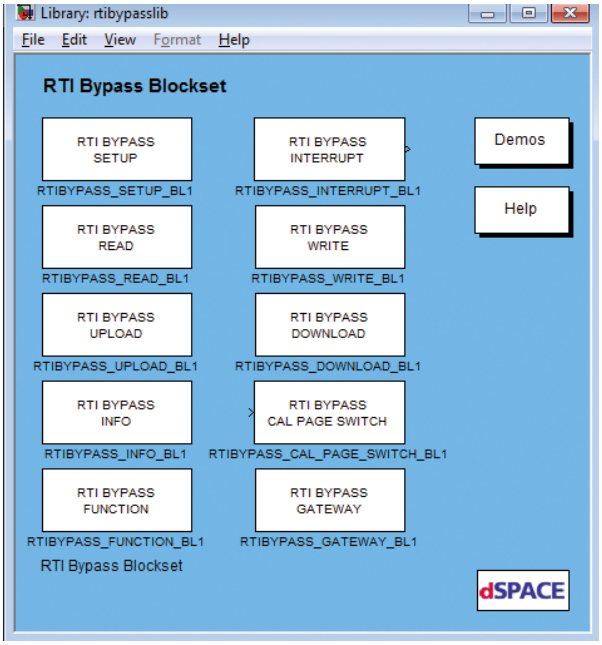
Name	Value
Name	Toggle
AddressHex	0xA0000B7C
FlagsHex	0x80

No.	Severity	Module	Time	Message
1	Info	GUI.Base	13:47:58.412	Application started!
2	Info	Project Man...	13:48:05.458	Loading project 'InternalBypassToggle' started ...
3	Info	Project Man...	13:48:07.327	Loading project 'InternalBypassToggle' finished.

Errors: 0 Warnings: 0 Information: 3

Real-Time Interface Bypass Blockset

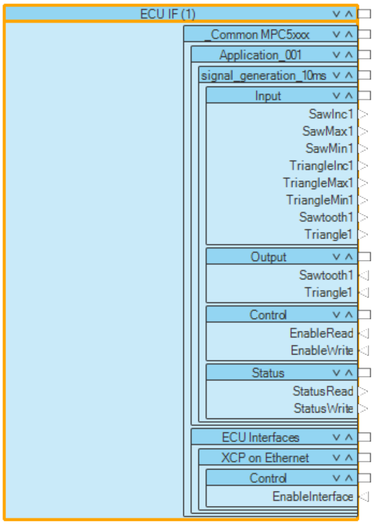
When working with PHS-based systems or MicroAutoBox II, the RTI Bypass Blockset provides a convenient blockset-based way of implementing a Simulink model that interfaces with a prepared ECU application. It lets you configure the access to the ECU services for reading and writing ECU variables and the corresponding execution synchronization. Using the optional On-Target Module also allows you to execute a Simulink model functionality directly on a target ECU and to also use the production code generator TargetLink.



ECU Interfacing with SCALEXIO Systems

ConfigurationDesk I/O Function

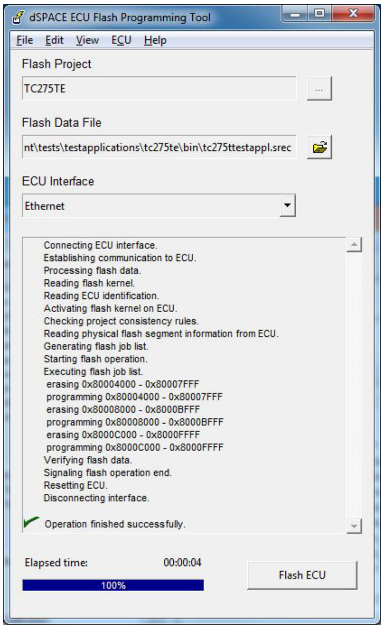
To use ECU Interfacing with SCALEXIO systems, ConfigurationDesk's ECU Interface Configuration I/O Function allows you to apply an ECU Access Configuration to your ConfigurationDesk model, as provided by the ECU Interface Manager.



Additional components of the ECU Interface Software

ECU Flash Programming Tool

In the context of ECU interfacing, new software versions have to be flashed to the ECU, e.g., after integrating an ECU service, or for on target bypassing. In case no in-house flashtool is available, dSPACE offers a generic substitute. With the ECU Flash Programming Tool, you can program ECUs via XCP on CAN/CAN FD and XCP on Ethernet (also supporting the dSPACE ECU Interface Hardware DCI-GSI2). The ECU Flash Programming Tool can be used as a stand-alone tool or it can be seamlessly accessed from within ControlDesk or RTI. Reliable ECU flash programming is guaranteed by flash data identification and consistency mechanisms. Very fast ECU flash programming can be implemented by using the DCI-GSI2. dSPACE provides consulting and engineering services to help you set up a flash programming solution for your ECU.



Flash Project: TC275TE

Flash Data File: nt\tests\testapplications\tc275te\bin\tc275ttestappl.srec

ECU Interface: Ethernet

```
Connecting ECU interface.
Establishing communication to ECU.
Processing flash data.
Reading flash kernel.
Reading ECU identification.
Activating flash kernel on ECU.
Checking project consistency rules.
Reading physical flash segment information from ECU.
Generating flash job list.
Starting flash operation.
Executing flash job list.
 erasing 0x80004000 - 0x80007FFF
 programming 0x80004000 - 0x80007FFF
 erasing 0x80008000 - 0x8000BFFF
 programming 0x80008000 - 0x8000BFFF
 erasing 0x8000C000 - 0x8000FFFF
 programming 0x8000C000 - 0x8000FFFF
Verifying flash data.
Signaling flash operation end.
Resetting ECU.
Disconnecting interface.

✓ Operation finished successfully.
```

Elapsed time: 00:00:04

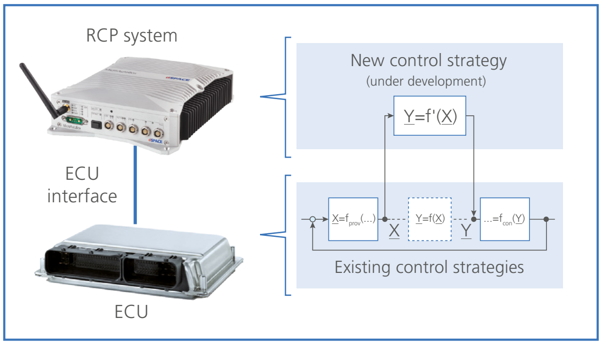
100%

Different Approaches and their Use Cases

ECU Interfacing is relevant in many different use cases. In the following some examples for the major approaches are shown.

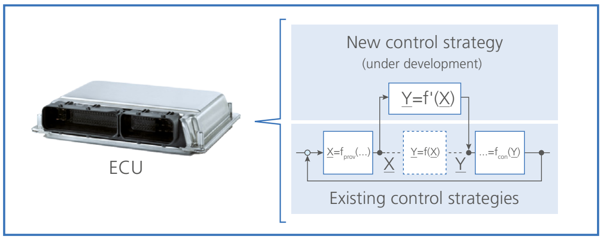
External Bypassing

External bypassing is an efficient approach for developing new control functions and optimizing existing controller strategies. The external bypass method uses a dedicated RCP system to execute new control functions synchronously to the original code on the target ECU. External bypassing allows even complex Simulink models to be executed as bypass functions, since there are almost no resource constraints, such as available RAM, ROM (flash memory) or processor performance. Furthermore, the RCP system provides additional I/O channels.



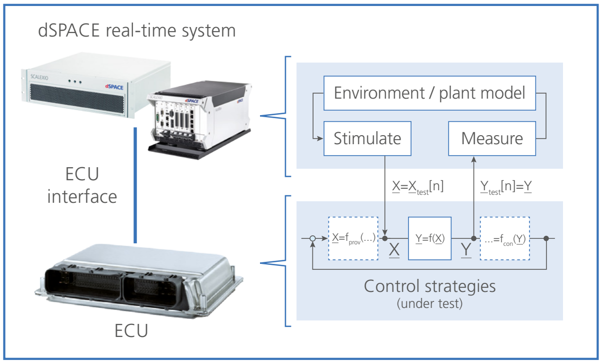
On-Target Bypassing

If an ECU provides all relevant I/O interfaces and sufficient free resources, function development can be performed directly on the ECU. This reduces development costs, because no additional hardware and extra wire harness is necessary. The new functions are executed directly on the target hardware, which means there are no communication latencies to external development hardware, and the functions can be integrated into very fast control loops. Using TargetLink as a code generator allows for a seamless transition to production and a more efficient use of the limited ECU resources. In addition, the resources required to run the function on the final production ECU can be identified very early on.



White-Box Testing, Function-in-the-Loop and Vehicle-in-the-Loop Simulation

In cases where it is difficult to provide or connect the real environment to the ECU for validation and test purposes, the ECU Interface tool chain can be used for connecting the ECU with the real-time system on system driver or application level. This gives you the ability to access the ECU-software-under-test conveniently from the real-time Simulink model. You can, for example, simulate values of inaccessible ECU sensors, such as the temperature, pressure or acceleration sensors, and feed them directly into the ECU on a driver level. This saves complex setups usually required for real stimulations (e.g., using climatic test chambers or motion platforms). This way of access also lets you react to the internal states of the ECU software during a running HIL test and thus specifically and immediately influence the test run. Moreover, isolated tests of the ECU software's sub-features can be performed purely by connecting the ECU hardware through the bypass interface to the dSPACE real-time system without the need of additional I/O wiring. (function-in-the-loop, white-box testing). In case of a vehicle-in-the-loop simulation where an ECU is tested in a real vehicle but under virtual conditions (e.g., virtual traffic), the ECU interface access allows you to simulate a virtual environment on a dSPACE real-time system that can be fed into the ECU in real-time.


dSPACE real-time system
ECU interface
ECU
Environment / plant model
Stimulate
Measure
X=X_test[n]
Y_test[n]=Y
X=f_prov(...)
X
Y=f(X)
Y
...=f_con(Y)
Control strategies
(under test)