On-target prototyping merges function and production development

Right on Target

With the dSPACE on-target prototyping solution, new functions can be integrated directly into existing electronic control unit (ECU) code and tried without having to perform complex software integration. This prototyping solution is extremely efficient regarding the resources of the control unit and allows a seamless transition to series production by using the production code generator dSPACE TargetLink.

When only the functionality of an existing production ECU needs to be extended, function development can be done directly on the ECU. If enough resources are still available and the existing I/O is sufficient, this can save the costs and effort for integrating and using additional prototyping hardware. Moreover, using the production code generator that is also used for the final ECU makes the code more efficient in terms of memory and execution time. On-target prototyping therefore automatically ensures that the resource limits of the ECU are observed. This reduces the project risk. Furthermore,
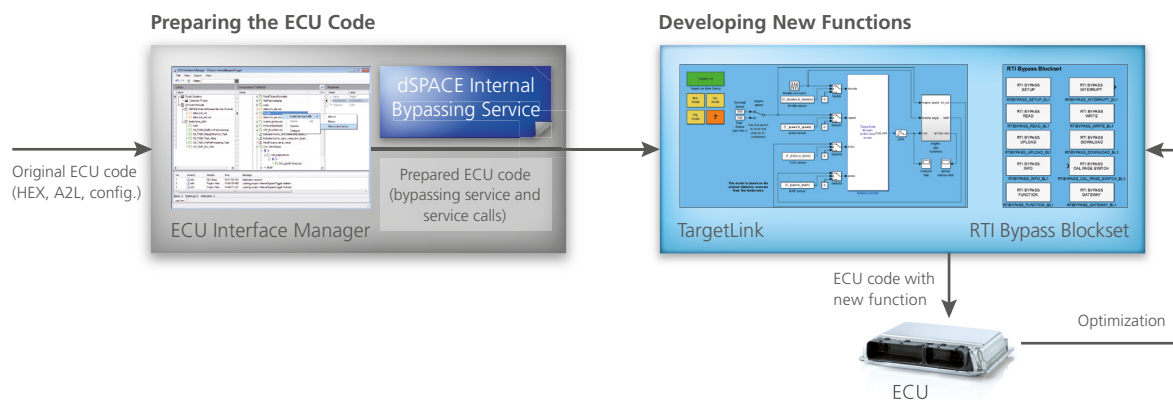
using the production code generator results in a seamless transition to the actual production development, including the convenience and quick iteration that is expected of proto-typing. The dSPACE on-target proto-typing solution with the production code generator TargetLink® can be used not only for developing new functions on the basis of Simulink®/TargetLink models. With minimal effort, these new functions can also be integrated as optimized code into existing ECU code for prototyping.

The dSPACE On-Target Prototyping Tool Chain

The core of the dSPACE on-target prototyping solution consists of the ECU Interface Manager, the dSPACE Internal Bypassing Service, the RTI Bypass Blockset and TargetLink (figure 1). The developers use the ECU Interface Manager to configure the bypass interfaces, which are required to integrate the new functions into the ECU code. Simulink/TargetLink are used to model the new functions. Afterwards, the new functions are connected to the existing ECU software with the RTI Bypass Blockset. This blockset connects the inports and outports of the function models with the interfaces that were

>>

Figure 1: In combination with the RTI Bypass Blockset, TargetLink implements new ECU functions in the ECU code, which was prepared with the ECU Interface Manager.
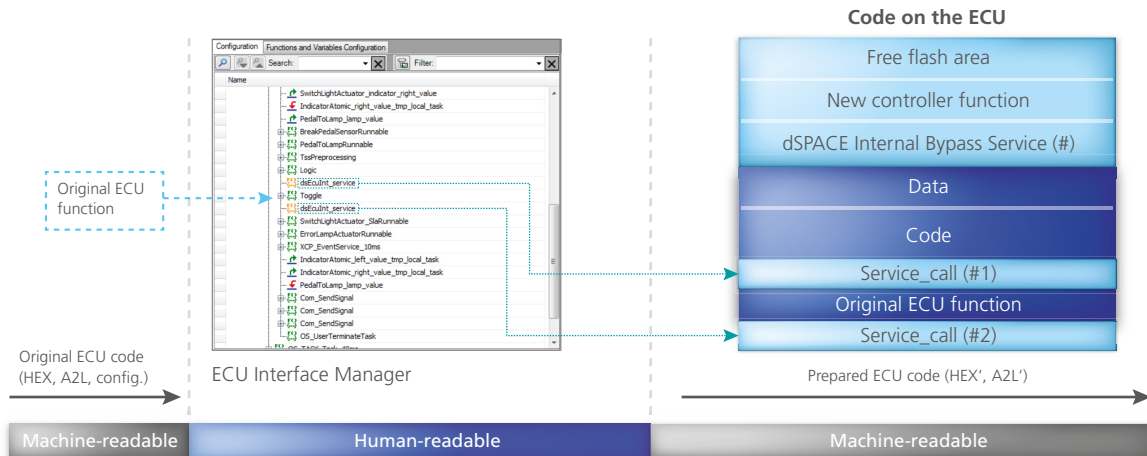

Preparing the ECU Code

Developing New Functions

Original ECU code
(HEX, A2L, config.)

dSPACE Internal Bypassing Service

Prepared ECU code
(bypassing service and service calls)

ECU Interface Manager

TargetLink

RTI Bypass Blockset

ECU code with new function

Optimization

ECU

Code on the ECU

Free flash area

New controller function

dSPACE Internal Bypass Service (#)

Data

Code

Service_call (#1)

Original ECU function

Service_call (#2)

Original ECU function

Original ECU code (HEX, A2L, config.)   ECU Interface Manager   Prepared ECU code (HEX', A2L')

Machine-readable   Human-readable   Machine-readable

Figure 2: To prepare the ECU code, the ECU Interface Manager analyzes the binary ECU software and displays it in a structured form, including the function names. Based on this, the bypassing service and the service calls are integrated into the new functions.

previously prepared in the existing binary code. Subsequently, the developers generate the production code with TargetLink and create the binary code for the ECU. The result is a revision of the ECU, which contains the newly developed functions. This revision is then transferred to the flash memory of the ECU. The developers can complete all these steps independently because no access to either the source code of the ECU software or a stand-alone build environment is required.

Integrating Bypassing Services Quickly and Easily
The ECU Interface Manager is an integral part of the on-target prototyping tool chain. The tool quickly integrates the bypassing services and the interfaces for the new functions into the ECU code, which is available as a binary file. No access to either the source code or the build environment of the ECU is required. The ECU supplier has to provide
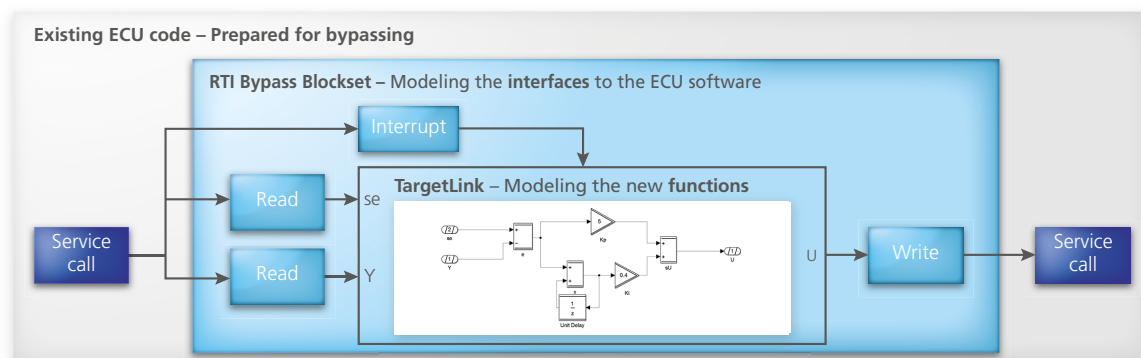
only some configuration information, e.g., free memory areas. No further iterations with the supplier are necessary, which saves project costs and time. The ECU Interface Manager, by contrast, uses a binary image of the ECU software to directly integrate the dSPACE Internal Bypassing Service into the ECU and to instrument the existing software in such a way that the new TargetLink functions can be integrated in all the required places. The ECU Interface Manager analyzes the program flow of the

> Developing new functions on the ECU fast and with optimized resource consumption

Figure 3: TargetLink and RTI Bypass Blockset for developing new functions.

Existing ECU code – Prepared for bypassing

RTI Bypass Blockset – Modeling the **interfaces** to the ECU software

Interrupt

TargetLink – Modeling the new **functions**

Service call

Read   se

Read   Y

U

Write

Service call

existing code for the supported processor families Infineon Tri-Core™, Renesas V850™ and NXP MPC 5xxx and provides the developers with the software structure and the relevant function names in a clear user interface for configuration (figure 2). In this user interface, the developers can specify, directly in the binary image, which interfaces are available during on-target prototyping. If functions are to be completely replaced, the developers can also specify to fully delete them from the ECU code and reuse the memory. In the end, the ECU Interface Manager generates a new ECU image that contains the bypassing service and the required service calls for the integration of the function that is to be developed.

Efficiently Using the ECU Resources

After the new ECU image has been prepared in the ECU Interface Manager, the new functions are developed with Simulink/TargetLink and the RTI Bypass Blockset. As a production code generator, TargetLink generates optimized ECU code and thus supports the optimal use of the limited resources. In addition, using TargetLink enables a seamless transition to production. The tool chain supports two different scenarios for modeling new functions:

1. *Developing Based on a TargetLink Model:* In this case, all TargetLink functions are directly available for the developers, who can make all the specifications that are desired for the final ECU code in the model as early as the function development phase.
2. *Developing Based on a Simulink Model:* TargetLink generates production code from the model and combines the code efficiency of TargetLink with reliable information about the used resources at maximum convenience.

In both cases, the universal RTI Bypass Blockset connects the modeled function and the ECU code (figure 3). The blockset offers flexible options for integrating the modeled function into the functional flow of the existing software. It provides access to variables of the existing software and makes it possible to call Simulink subsystems synchronously with the processing of the original ECU application.
After configuring the connection to the ECU software, the developers can start the automatic build process for new functions at the click of a button. The build process automatically integrates the function into the ECU image. For this, it even uses fragmented memory areas to optimally use available resources.

Optionally, it starts the flash process and thus directly transfers the newly created ECU image to the ECU without the need for manual steps. Access to the measurement and calibration parameters is provided via the existing ECU interfaces, as usual. ∎

Benefits and Future Innovation

The combination of the ECU Interface Manager, RTI Bypass Blockset and TargetLink reconciles the fast iterations of rapid control prototyping with the high demands for efficiency and configurable production code. The existing production ECU can be conveniently used as prototyping hardware. As a result, resource consumption is kept under control, and due to the continued use of the function models with TargetLink, a seamless transition to production development is possible.
The binary-code-based integration of service calls for bypassing with the ECU Interface Manager saves tedious integration loops that involve the ECU supplier. Thus, prototyping can start immediately. For future dSPACE Releases, the new feature for virtual bypassing with TargetLink and dSPACE VEOS® on a host PC will increase productivity even further.

Table 1: The tool chain for close-to-production on-target prototyping.

dSPACE Tool	Task
ECU Interface Manager	Intuitive tool for quickly integrating service calls for bypassing directly into existing ECU code
dSPACE Internal Bypassing Service	ECU service for extending existing ECU code with service calls for bypassing
TargetLink®	Software system that automatically generates production code (C code) directly from the graphical MATLAB®/Simulink®/Stateflow® development environment
RTI Bypass Blockset	Simulink blockset for easily connecting new functions with existing ECU code
Target-specific compiler (third-party product: HighTec compiler)	Transferring C code into object code for the processor families Infineon TriCore™, Renesas V850™, and NXP MPC5xxx