



A winning Hand

TargetLink 3.1 brings impressive features into play

TargetLink 3.1 has major extensions to the code generator's core functionality and to AUTOSAR support, as well as improved integration in MATLAB®/Simulink®. Enhanced usability is also on the table.

A solid player, now perfected even further – TargetLink Version 3.1 makes production code generation more attractive and more powerful.

Code Generation Straight from the dSPACE Data Dictionary

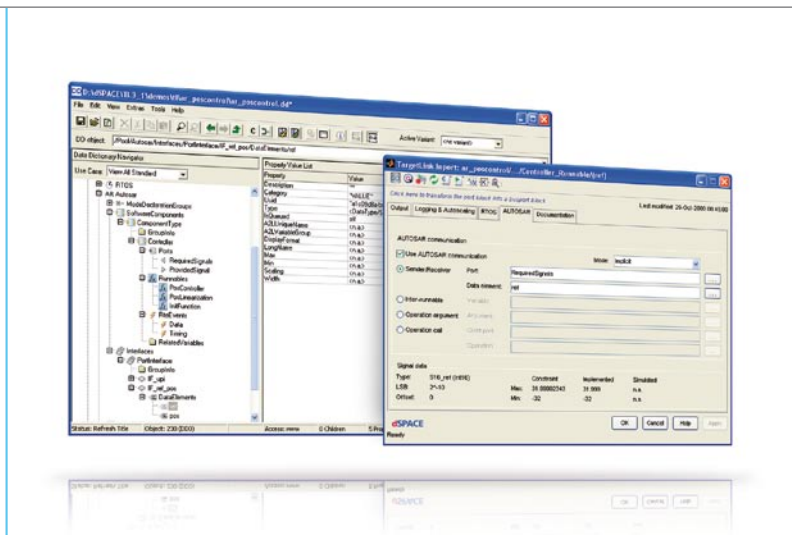
One major innovation in TargetLink 3.1 is that code can now be generated not only from models, but also directly from the central data container, the dSPACE Data Dictionary. This is an immense advantage for software integration and software

integration testing. Variables that are handled by more than one developer, such as interface variables, measurement and calibration variables, or legacy code parameters, can be defined in the dSPACE Data Dictionary and assigned to individual modules. Code and A2L files (ASAP2) are then generated for these variables directly from the dSPACE Data Dictionary, independently of any specific modules. For example, all the calibration parameters of an entire project, including the legacy

variables, can be managed in the dSPACE Data Dictionary and generated in one single C file and one single A2L file. In addition, software integration tests are much easier because of TargetLink's new model referencing and incremental code generation features.

Vectorized Code with Variable Vector Widths

With TargetLink 3.1, users have even more flexibility when generating code for vector signals. Code for vectors can be generated with vector widths that are not defined by a fixed number but determined by a macro. This means that developers can reuse the same code for different vector widths (for example, for 4-, 6- and 8-cylinder engines), which considerably reduces the effort involved in code reviews and tests.



The new TargetLink AUTOSAR blockset: Simpler model migration and seamless integration into the Simulink world.

Traceability from Requirements to Code

TargetLink 3.1 makes it easier to track requirements through to the generated code, which in turn simplifies process-compliant workflows that follow standards such as IEC 61508 or ISO 26262 with TargetLink. If requirements are linked to models, TargetLink inserts them into the generated code as comments. In addition, the automatically generated documentation clearly tells which requirement was implemented in which model parts to ensure a completely transparent development process.

Native Support for Bit Operation Blocks

The extended TargetLink block library now gives users native support for bit operations such as bit set, bit clear, bitwise operations, extract bits and arithmetic bit shifts. The new bit operation blocks not only provide TargetLink's typical user-friendly signal specification and visualization, but also lead to highly efficient code with the help of interblock optimization.

Extensive New AUTOSAR Support Features

The new TargetLink AUTOSAR blockset is directly integrated into the regular TargetLink blockset. This not only makes it easier to migrate

from conventional TargetLink models to AUTOSAR, but integration into the Simulink world is also seamless. When the TargetLink AUTOSAR Migration Tool is used, conventional TargetLink models can be migrated to AUTOSAR at the push of a button and then used for generating both conventional and AUTOSAR-compliant code, dramatically reducing model maintenance work. TargetLink now supports AUTOSAR Standard 3.1 as well as numerous other AUTOSAR features such as client-server communication for complex data types, signal acknowledgement and signal invalidation in data transmission, and per instance memories. TargetLink's interaction with dSPACE SystemDesk and other architecture tools has also been further optimized to enable a seamless, iterative AUTOSAR development process.

In TargetLink 3.1, the same code can be reused for different vector widths, because the widths are defined flexibly via a macro.

```
#define NumOfCyl 4

....

Float64 Sal_U[NumOfCyl];
Float64 Sal_Y[NumOfCyl];

....

for (Aux_S32 = 0; Aux_S32 < NumOfCyl; Aux_S32++)
{
    Sal_U[Aux_S32] = (Sal_REF[Aux_S32] * ((Float64) P_Sal_Kp[Aux_S32] *
    X_Sal_Unit_Delay[Aux_S32] + (1 - P_Sal_Kp[Aux_S32]) * Sal_U[Aux_S32]) *
    /* Unit delay: picontroller/Unit Delay [0.001] * NumOfCyl-1 */
    X_Sal_Unit_Delay[Aux_S32] = Sal_Y[Aux_S32];
}
}
```

Enhanced Usability and MATLAB/Simulink Integration

Day-to-day work with TargetLink is even easier with Version 3.1. For example, the Data Dictionary Manager now has its own message browser to display messages. User-configurable (context) menus can be inserted in the Data Dictionary Manager and underlaid with user scripts. Different sets of code generation options can be stored in the dSPACE Data Dictionary in a uniform manner, making it easier for developers to exchange them. TargetLink 3.1 also provides more advanced dialogs to link a model with the dSPACE Data Dictionary, plus enhanced Simulink integration of TargetLink demo models and TargetLink-specific menus.

Extensions to the TargetLink Simulation Module

The TargetLink Simulation Module (TSM) in TargetLink 3.1 now also supports the execution of processor-in-the-loop simulations (PIL) for the Infineon TriCore TC 1767 controller in conjunction with Tasking compilers. ■