Automatic Code Generation Technology

For Embedded Control Applications

The MathWorks provides a complete, end-to-end solution for designing and implementing embedded control systems. The MathWorks model-based design tools make all phases of the development process easier — from concept creation through software deployment on embedded microprocessors. The high level of integration of MATLAB®, Simulink®, Real-Time Workshop®, and Real-Time Workshop® Embedded Coder enables you to move smoothly through modeling, analysis, simulation, code generation, prototyping, testing, and deployment. Real-Time Workshop provides the code generation infrastructure for applications such as rapid prototyping, hardware-in-the-loop (HIL) simulation, and host-based simulation acceleration. Real-Time Workshop Embedded Coder extends these capabilities with advanced features for memory optimization and production system deployment.

Real-Time Workshop

Use Real-Time Workshop to rapidly explore and test new concepts and designs for discrete, continuous, or hybrid systems. Real-Time Workshop is a highly efficient C code generator that is used by system designers who require rapid prototyping, hardware-in-the-loop simulation, stand-alone (non-real-time) simulation, or generic algorithm deployment capability. In these phases of the design process, developers require robust analysis, testing, and debugging capabilities, and can postpone considering embedded target implementation issues, such as RAM constraints, code configuration, component testing, and code traceability. Features in Real-Time Workshop that aid the system designer at these stages include:

- Static and dynamic memory allocation. Static memory allocation provides maximum efficiency. Dynamic memory allocation enables multi-instanced (reentrant) code.
- Single-rate, multirate, and asynchronous rate model support.
- Automatic or manual subsystem code reuse.
- Singletasking and multitasking support for multirate deployment, including S-functions.
- Data type support for single and double precision; signed and unsigned 8-, 16-, and 32-bit integers; Boolean; and extended fixed-point types.
- Simulink data object support for defining signals, parameters, states, and data store memory.
- External Mode, enabling Simulink to communicate with the generated code running on the target.
- MAT-file logging for validating results of the generated code in MATLAB.
- Well-defined C interfaces for monitoring signals and tuning parameters with MATLAB and Simulink.
- ASAP2 data export capability. ASAP2 is a data definition standard proposed by the Association for Standardization of Automation and Measuring Systems for interfacing to commercial automotive calibration standards.
- S-Function target for generating portions of your models as CMEX S-functions. Use the S-functions for accelerating simulations, protecting intellectual property, creating reusable components, and generating code incrementally.
- Rapid simulation (RSIM) target for generating fast, stand-alone (non-real-time) simulations using variable-step or fixed-step solvers.

Real-Time Workshop is a framework for creating custom, highly integrated code generation solutions and add-on products. It lets you quickly prototype applications that do not require deployment features for addressing the special needs and constraints of embedded discrete time targets.

Real-Time Workshop Embedded Coder

To support the development and deployment of discrete-time applications on embedded targets, The MathWorks provides Real-Time Workshop Embedded Coder, which extends the capabilities of Real-Time Workshop to address RAM/ROM/CPU constraints, code configuration, component testing, code traceability, and other target-specific considerations.

The following options are available exclusively through Real-Time Workshop Embedded Coder:

- A simplified calling interface that reduces overhead and lets you easily incorporate the generated code into handwritten application code. Configuration options allow you to communicate data to your application code via function arguments or global data structures.
- Automatic generation of an example main program detailing precisely how to deploy the code generated for a model with or without an operating system.
- Automatic generation of a deterministic multirate scheduler for single-tasking and multitasking environments (further simplifying code deployment).
- Single-instance and multi-instance (reentrant) code generation, both using static memory allocation.
- Streamlined data structures, reducing RAM requirements.
- Combined output and update functions, significantly reducing code size.
- Configuration options to optimize data initialization, reducing ROM requirements.
- Support for ISO-C, improving single-precision math performance.
- Software-in-the-loop code validation, simplifying testing the generated code. The generated code is compiled and run on the host system in a closed-loop simulation inside Simulink.
- Custom storage classes, giving you precise control over the symbolic representation of data. This feature lets you easily interface data used by the generated code to virtually any class of structured or unstructured data required by your existing code base. Several predefined storage classes are provided to work with signals, parameters, states, and data store memory data.
- Extended ASAP2 data export. Custom storage classes, used in conjunction with the Real-Time Workshop ASAP2 feature, allow the export of arbitrarily complex data structures in ASAP2 format.

Code Formats

A code format specifies the overall framework of the generated code. Depending on your application's constraints and requirements, you can select one of several code formats. Real-Time Workshop includes the generic real-time (GRT), generic real-time malloc (GRT-malloc), and S-function code formats. Each code format supports discrete, continuous, and hybrid systems. Real-Time Workshop Embedded Coder adds the Embedded Real-Time (ERT) code format. The ERT code format is aimed at production embedded targets, which do not need to support continuous-time blocks. By eliminating this requirement, the ERT code format creates code with simpler, smaller data structures and reduced calling overhead. It handles real-time, real-world demands better than the GRT code format.

Any of these code formats can be deployed on real-time or non-real-time targets. The ERT code format, however, is the most suitable for deployment on embedded real-time targets.

Key Features

Key Product Features	Real-Time Workshop	Real-Time Workshop-Embedded Coder
Custom storage classes		X
HTML report generation	Basic	Extended
Options for streamlining model initialization code		X
Software-in-the-loop code validation		X
ASAP2 data export	Basic	Extended
Target-independent optimizations	X	X
Target-specific optimizations		X
Simplified code interface		X
Lightweight code framework		X
Code reuse for nonvirtual subsystems	X	X
Support for ISO-C		X
Supported Code Formats		
Embedded real-time		X
Generic real-time (static memory allocation)	X	X
Generic real-time malloc (dynamic memory allocation)	X	X
S-function	X	X

Optimizations

The performance of generated code is important for all applications. Optimizations that improve performance, such as redundant operation, conditional branch execution, code reuse, dead path elimination, and inlined functions and constants, are desirable in any target environment and are built into Real-Time Workshop. Optimizations that reduce memory usage or are target-specific are generally not necessary for all applications, and therefore are provided by Real-Time Workshop Embedded Coder and related Embedded Targets.

Rapid Simulation

The Rapid Simulation (RSIM) target generates fast, stand-alone simulations for models using variable or fixed-step solvers. RSIM supports modification of input signal data and model parameters by reading data from a MAT-file, without recompiling the generated code. This is particularly useful for batch and Monte Carlo simulation runs. RSIM is included with Real-Time Workshop.

Product Family Overview

MathWorks control design automation products are structured to maximize flexibility and minimize the cost of dealing with the number and variety of viable targets in the market. Several MathWorks products and components within them address the growing number of rapid prototyping and real-time applications. Individual MathWorks products and components work together to address specific application requirements and constraints. Figure 1 illustrates the relationship between the main products and components used to automatically generate code from Simulink models.

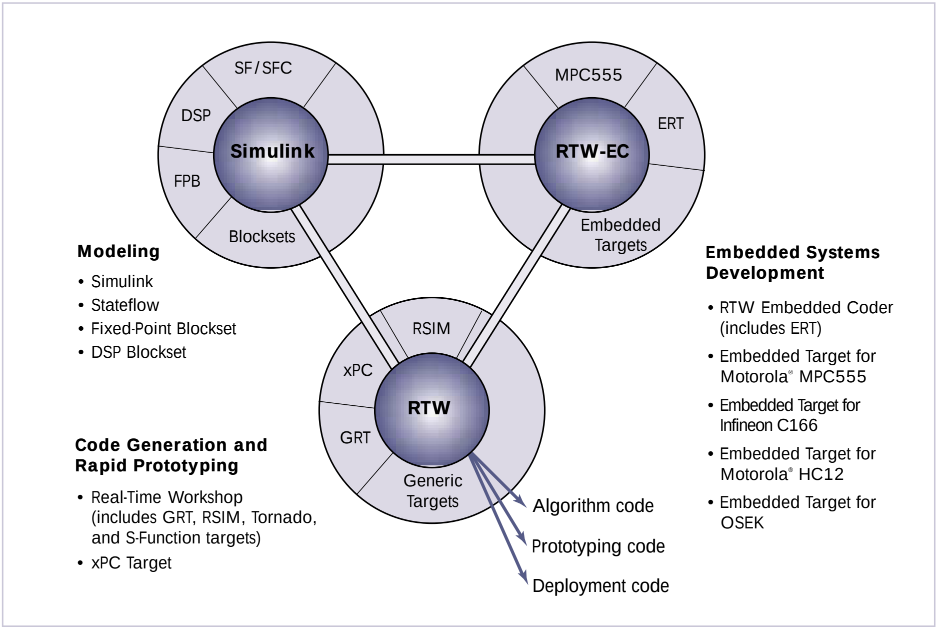
SF / SFC
DSP
Simulink
FPB
Blocksets
MPC555
ERT
RTW-EC
Embedded Targets
RSIM
xPC
RTW
GRT
Generic Targets
Algorithm code
Prototyping code
Deployment code
Modeling
• Simulink
• Stateflow
• Fixed-Point Blockset
• DSP Blockset
Code Generation and Rapid Prototyping
• Real-Time Workshop (includes GRT, RSIM, Tornado, and S-Function targets)
• xPC Target
Embedded Systems Development
• RTW Embedded Coder (includes ERT)
• Embedded Target for Motorola® MPC555
• Embedded Target for Infineon C166
• Embedded Target for Motorola® HC12
• Embedded Target for OSEK

Figure 1

Figure 1 shows that Real-Time Workshop is central to MathWorks code generation, has a direct link to Simulink, and is used directly for rapid prototyping, stand-alone simulation, and generic algorithm deployment. For embedded systems development, the link between Simulink and code generation by Real-Time Workshop is through Real-Time Workshop Embedded Coder, where the additional capabilities for embedded systems are added.

Notes for DSP and Communications Applications

You can generate ANSI C code from DSP Blockset blocks using either Real-Time Workshop or Real-Time Workshop Embedded Coder. All DSP Blockset blocks support the following code generation targets:

- Generic real-time (GRT)— in Real-Time Workshop
- Embedded real-time (ERT)—in Real-Time Workshop Embedded Coder

Real-Time Workshop Embedded Coder features are not accessible to or used by the Embedded Target for TI C6000™ DSP, since many Real-Time Workshop Embedded Coder capabilities are not currently compatible with the TI Code Composer Studio® IDE.

Summary

When selecting MathWorks code generation products for your applications, consider these two key points:

- If your main concerns are robust analysis, testing, and debugging capabilities (which is the case for rapid prototyping and hardware-in-the-loop simulation) then you need Real-Time Workshop.
- If you have target-specific requirements, such as RAM constraints, code configuration, component testing, and code traceability, then you need Real-Time Workshop Embedded Coder in addition to Real-Time Workshop.

The MathWorks