Latest Features in Embedded Coder

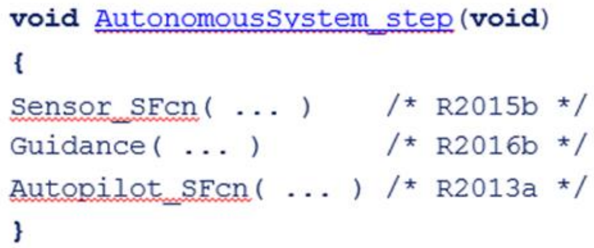
March 2017

R2017a

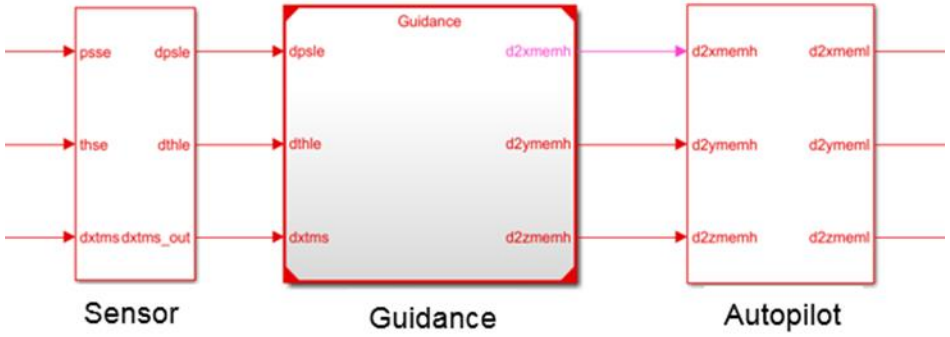
Cross-Release Code Integration

Reuse model reference code generated from previous releases

- You can integrate exported component code that uses the model reference code interface
- Previously, the cross-release integration workflow supported only component code that used the standalone code interface



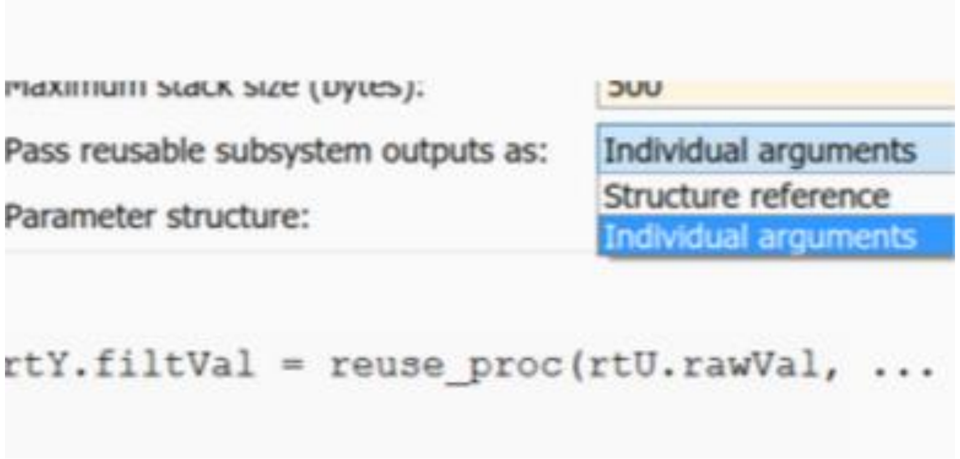
```
void AutonomousSystem_step(void)
{
Sensor_SFcn( ... )      /* R2015b */
Guidance( ... )         /* R2016b */
Autopilot_SFcn( ... )  /* R2013a */
}
```



Function Interface

Return nonvoid type for scalar output of reusable functions

- Previously, the reusable functions had a return type of void
- In R2017a, reusable functions can return a nonvoid type
- Conserves RAM consumption because the generated code does not contain a global variable to hold the output parameter value



Maximum stack size (bytes): 500

Pass reusable subsystem outputs as: Individual arguments

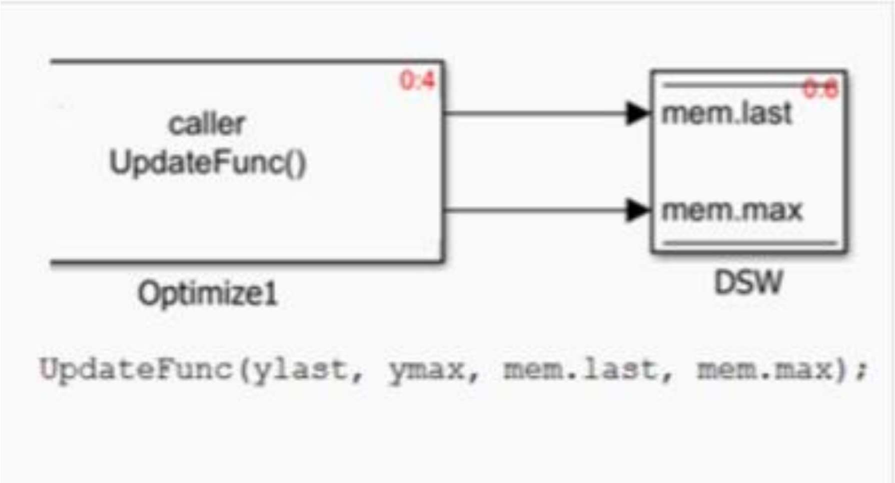
Parameter structure:

```
rtY.filtVal = reuse_proc(rtU.rawVal, ...
```

Data Copy Reduction

Generate fewer data copies and use less RAM for buses, data stores, and model blocks

- Generated code contains less temporary variables and associated data copies for modeling patterns involving
  - Bus Assignment
  - Data Store Read and Write
  - Model blocks
- These optimizations conserve RAM usage and improve code execution speed



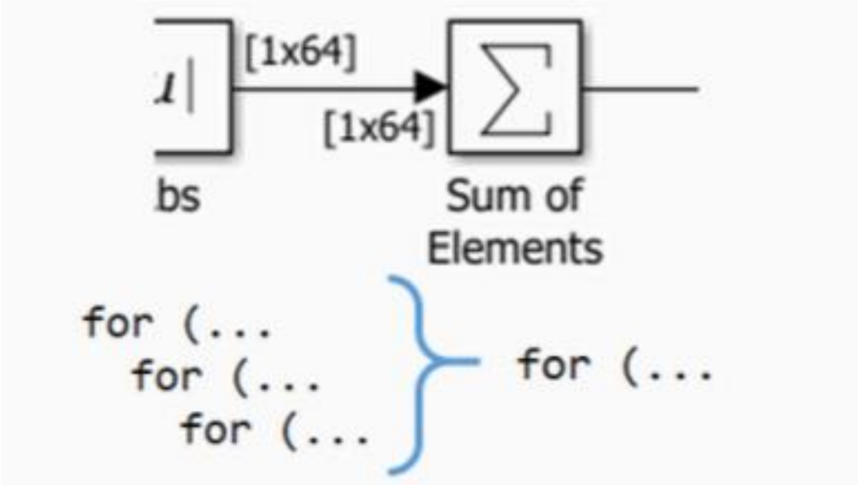
```
UpdateFunc(ylast, ymax, mem.last, mem.max);
```

MathWorks R2017a

Code Efficiency

Improve loop fusion for Sum of Elements blocks and generate less code for temporal logic in Stateflow

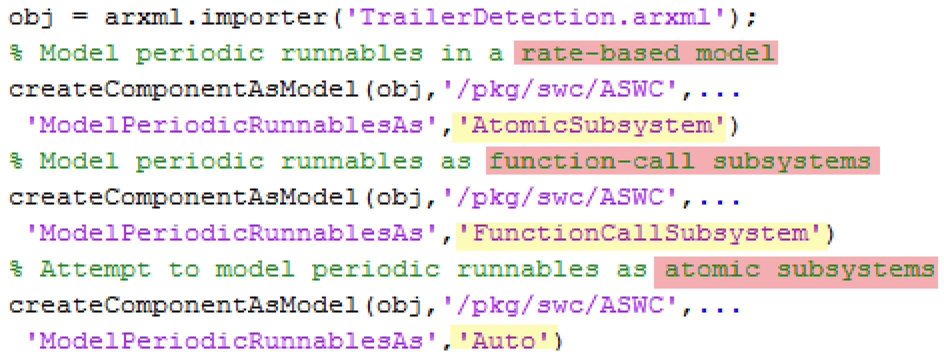
- Code generator can fuse more for loops involving Sum of Elements blocks.
- This optimization conserves ROM consumption and improves code execution speed.



AUTOSAR `arxml` File Import

Flexibly model imported periodic, asynchronous, and initialization runnables

- AUTOSAR `arxml` importer supports all modeling styles
  - Import periodic and asynchronous runnables in a JMAAB type beta modeling configuration
  - Import an initialize runnable, which the importer now represents with a Simulink Initialize Function block

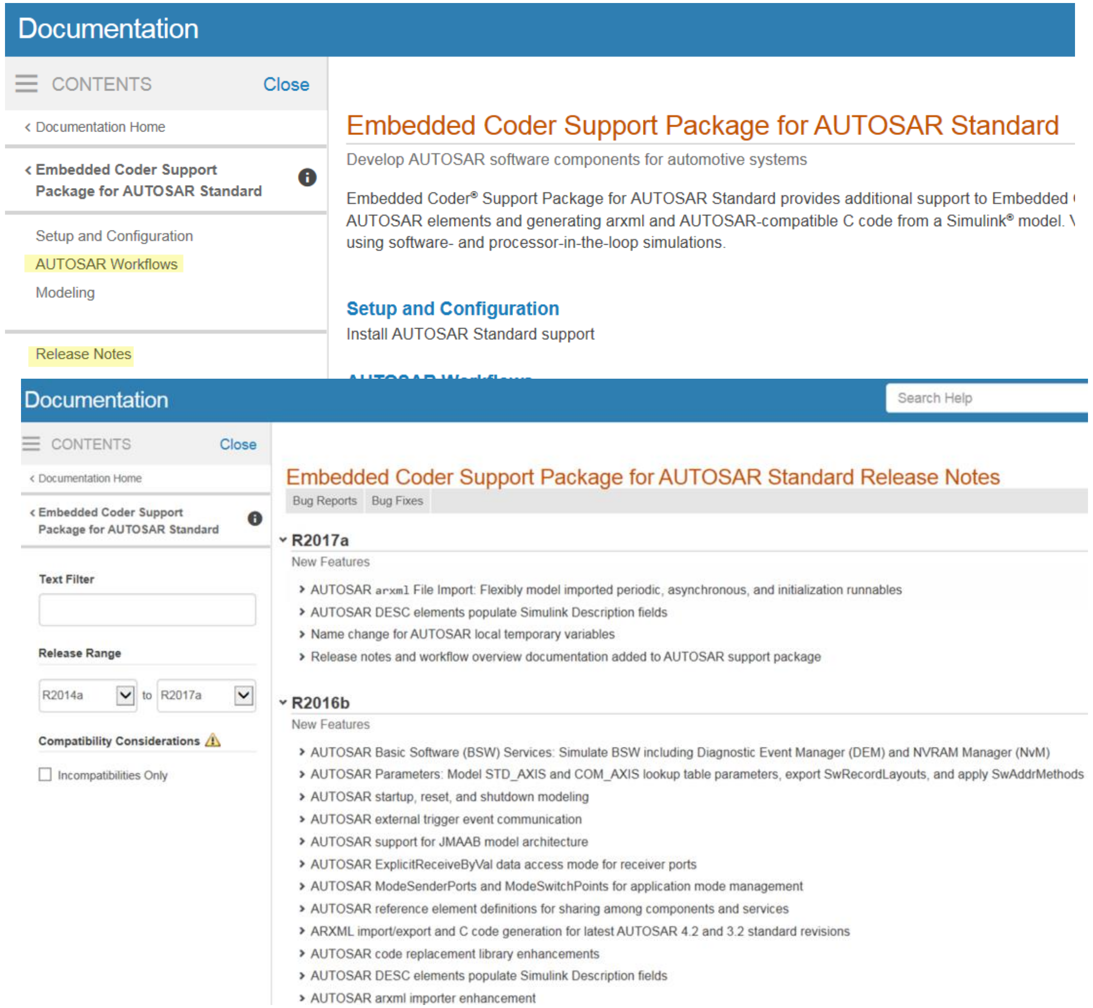


```
obj = arxml.importer('TrailerDetection.arxml');
% Model periodic runnables in a rate-based model
createComponentAsModel(obj,'/pkg/swc/ASWC',...
 'ModelPeriodicRunnablesAs','AtomicSubsystem')
% Model periodic runnables as function-call subsystems
createComponentAsModel(obj,'/pkg/swc/ASWC',...
 'ModelPeriodicRunnablesAs','FunctionCallSubsystem')
% Attempt to model periodic runnables as atomic subsystems
createComponentAsModel(obj,'/pkg/swc/ASWC',...
 'ModelPeriodicRunnablesAs','Auto')
```

Separate Release Notes for AUTOSAR Support Package

Added workflow overview documentation

- Release notes describe AUTOSAR support changes from the current release back through R2014b



Documentation

Embedded Coder Support Package for AUTOSAR Standard

Develop AUTOSAR software components for automotive systems

Embedded Coder® Support Package for AUTOSAR Standard provides additional support to Embedded AUTOSAR elements and generating arxml and AUTOSAR-compatible C code from a Simulink® model. using software- and processor-in-the-loop simulations.

Setup and Configuration

Install AUTOSAR Standard support

AUTOSAR Workflows

Documentation

Embedded Coder Support Package for AUTOSAR Standard Release Notes

R2017a

New Features

- AUTOSAR `arxml` File Import: Flexibly model imported periodic, asynchronous, and initialization runnables
- AUTOSAR DESC elements populate Simulink Description fields
- Name change for AUTOSAR local temporary variables
- Release notes and workflow overview documentation added to AUTOSAR support package

R2016b

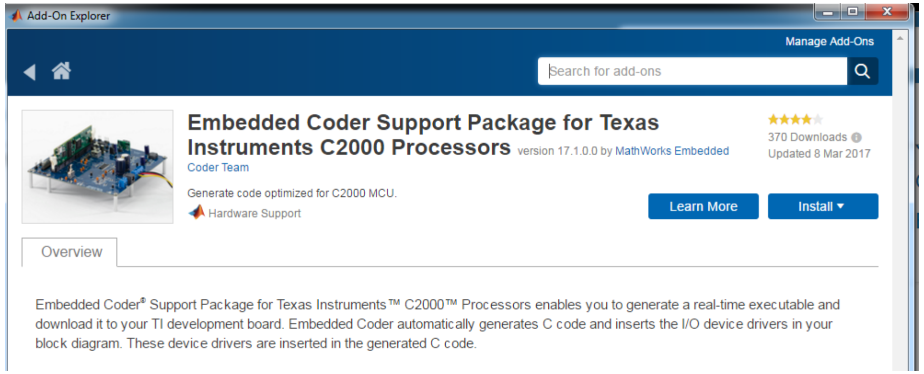
New Features

- AUTOSAR Basic Software (BSW) Services: Simulate BSW including Diagnostic Event Manager (DEM) and NVRAM Manager (NvM)
- AUTOSAR Parameters: Model STD_AXIS and COM_AXIS lookup table parameters, export SwRecordLayouts, and apply SwAddrMethods
- AUTOSAR startup, reset, and shutdown modeling
- AUTOSAR external trigger event communication
- AUTOSAR support for JMAAB model architecture
- AUTOSAR ExplicitReceiveByVal data access mode for receiver ports
- AUTOSAR ModeSenderPorts and ModeSwitchPoints for application mode management
- AUTOSAR reference element definitions for sharing among components and services
- ARXML import/export and C code generation for latest AUTOSAR 4.2 and 3.2 standard revisions
- AUTOSAR code replacement library enhancements
- AUTOSAR DESC elements populate Simulink Description fields
- AUTOSAR arxml importer enhancement

TI Code Composer Studio (CCS)

Generate projects for CCS versions 5 and 6 with Embedded Coder Target for TI C2000

- Code Composer Studio project is generated when you build Simulink models for TI C2000 targets with CCS v5 or v6 toolchains,
- You can use this project for debugging the generated code



Embedded Coder Support Package for Texas Instruments C2000 Processors version 17.1.0.0 by MathWorks Embedded Coder Team

★★★★☆
370 Downloads
Updated 8 Mar 2017

Generate code optimized for C2000 MCU.

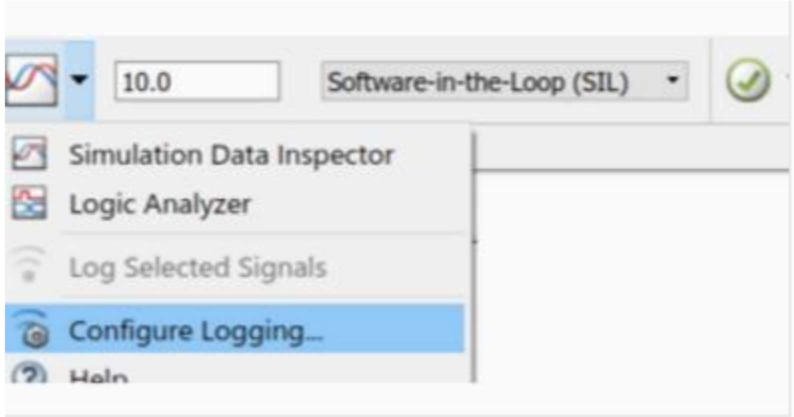
Hardware Support

Embedded Coder® Support Package for Texas Instruments™ C2000™ Processors enables you to generate a real-time executable and download it to your TI development board. Embedded Coder automatically generates C code and inserts the I/O device drivers in your block diagram. These device drivers are inserted in the generated C code.

SIL and PIL Testing

Log signals inside exported functions and stream signals to Simulation Data Inspector during simulation

- Enable internal signal logging for a top-model or model block software-in-the-loop (SIL) or processor-in-the-loop (PIL) simulation
- You can also
  - Log signals inside export-function models
  - Stream the logged signals to the Simulation Data Inspector, to observe the signals during the SIL or PIL simulation


10.0
Software-in-the-Loop (SIL)
Simulation Data Inspector
Logic Analyzer
Log Selected Signals
Configure Logging...
Help