Unified Modeling Architecture

• Objective:
  To create a controller model which can be used for both simulation and production code generation while supporting periodic and asynchronous functions.

• Approach:
  – Model the periodic function as a separate subsystem.
  – Model asynchronous function as a function call subsystem.
  – Depending on the system these function call subsystem can be a complex function callback that pend on some low priority events or simple callback for the high priority hardware interrupts.
Modeling and Code Generation Recommendations

• When using function calls to implement the callback for hardware interrupts, all the rules applicable to ISR will apply.
• When the function output is controlled by both periodic and asynchronous functions use the data stores to merge the outputs.
• Place the function call subsystem inside a triggered subsystem to be able to simulate the asynchronous ISR behavior during simulation.
• Set the code generation properties to generate code for function call subsystem as a separate function in a separate file.
• The generated code has:
  – step function which has all the periodic functions calls.
  – function definition for the function call subsystem.
• Custom block in the function call generator to avoid the function getting called in the step function.
Advantages Of This Approach

- A controller model that incorporates the asynchronous external event handling mechanism.
- Same model is used for simulations and code generation.
- External event handler callback is built inside the model.
- This callback can be tested through simulation.
- Code generated for this callbacks can be used as such in the framework software while integration.