

Description

This example is focusing on the integration of C code in MATLAB and Simulink. The goal is to show C code integration and C code generation out of MATLAB and Simulink algorithms for a parking meter application. A C# interface is used for the front-end of the parking meter and it can be controlled with MATLAB and Simulink through a Windows shared memory interface. There are two implementations of the algorithm controlling the C# interface; one in MATLAB only and one in Simulink. In both cases, the goal is first to be able to simulate the entire system by integrating the low-level C interface drivers and then to generate C code out of the algorithm and create a standalone executable automatically.











Configuration

In all following implementations folders (MATLAB, Simulink and Stateflow), there is a full Project that can be opened by double-clicking on the project's file "ParkingMeter.prj". This will load and setup the project automatically in MATLAB. Then, in the "PROJECT SHORTCUTS" tab on top, the list of all the main modules of the project are shown. The contents of the "PROJECT SHORTCUTS" tab depends on the selected implementation.

The "Readme.txt" file provides the needed steps to do to properly run the application.

MATLAB












The "PROJECT SHORTCUTS" content for the MATLAB implementation is the following:

HOME	PLOTS	APPS	SHORTCUTS	PROJECT	PROJECT SHORTCUTS
 New Shortcut  Organize Groups MANAGE	 Setup  Shutdown 0-PROJECT	 Main Function 1-ALGORITHM	 Low Level  Quick 2-TESTING	 Run C Code 3-CODEGEN	 Debug  Release 4-STANDALONE

1. ALGORITHM : Entry point of the application (main MATLAB file)
2. TESTING : MATLAB scripts to test the interaction with the C# interface
3. CODEGEN : Generate C code out of the application and execute it
4. STANDALONE : Executable binaries ready to run

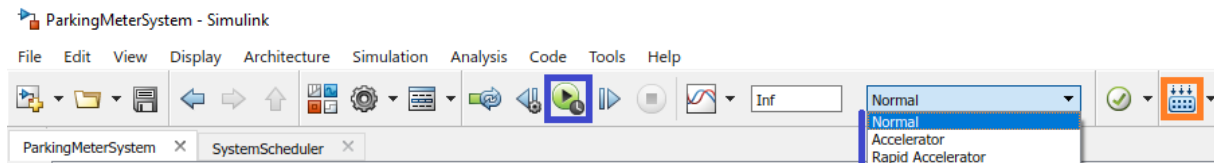
Simulink

The "PROJECT SHORTCUTS" content for the Simulink implementation is the following:

HOME		PLOTS	APPS	SHORTCUTS	PROJECT	PROJECT SHORTCUTS	
			 <u>Main System</u>		 Read Register	 Components	 Low Level
New Shortcut	Organize Groups		 Scheduler		 Write Register	 Interface	 System
MANAGE		1-MODELS	2-COMPONENTS	3-DATA	4-INTERFACE	5-LIBRARY	6-STANDALONE

1. MODELS : Entry point of the application for the **Model** approach
2. COMPONENTS: Entry point of the application for the **System** components approach
3. DATA : Parameters and buses of the application
4. INTERFACE : Low-level C drivers wrapped in Simulink to access the C# interface
5. LIBRARY : List of modules/components that can be used for the implementation
6. STANDALONE : Executable binaries ready to run

Once one of the two implementations has been loaded, it is possible to run the simulation in “**Normal**”, “**Accelerator**” or “**Rapid Accelerator**” modes. It is also possible to build the model as a standalone executable by clicking on the “**Build**” button has shown below:



Stateflow

The “PROJECT SHORTCUTS” content for the Stateflow implementation is the following:

HOME	PLOTS	APPS	SHORTCUTS	PROJECT	PROJECT SHORTCUTS	
New Shortcut Organize Groups MANAGE	Setup Shutdown 0-PROJECT	Main Function 1-ALGORITHM	Scheduler 2-SCHEDULING	Low Level Quick 3-TESTING	Run C Code 4-CODEGEN	Debug Release 5-STANDALONE

1. ALGORITHM : Entry point of the application (main MATLAB file)
2. SCHEDULING : Stateflow chart of the application’s scheduler
3. TESTING : MATLAB scripts to test the interaction with the C# interface
4. CODEGEN : Generate C code out of the application and execute it
5. STANDALONE : Executable binaries ready to run

Notes

- The installation of a Microsoft Visual C/C++ compiler is needed to compile and build the code, and its setup has to be done in MATLAB with the command:
`>> mex -setup`
- There is also a “C_project” folder that contains a Visual Studio 2019 project to build the standalone executable of the application in each implementation
- The C# interface has no intelligence at all; this means that if you simply run the file “ParkingMeterGUI.exe” without MATLAB, Simulink or the standalone executables, nothing will happen in the interface and that is normal. To be able to control it, you need to run the control application algorithm implemented in MATLAB or Simulink as explained in the above sections