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.

MATLAB

In this folder, there is all the code that can be edited and run directly from within MATLAB. Its "Readme.txt" provides the needed steps to do to properly run the application. The main files of the project are:

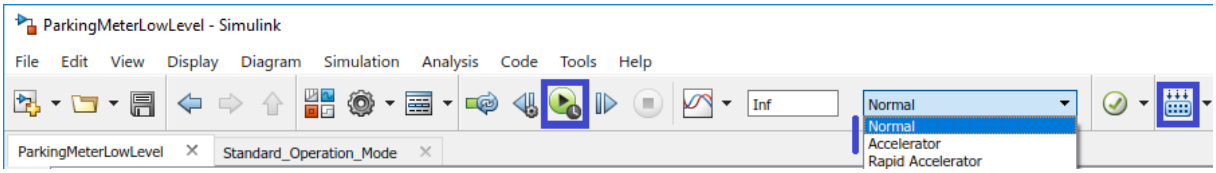
- "ParkingMeterLowLevel.m" in the "Functions" folder that runs the main application
- the low-level C interface drivers in the "Interface_C_files" and "MEX_functions" folders

There is also a "C_project" folder that contains a Visual Studio 2017 project to build a standalone executable of the application.

Simulink

In this folder, there is a full Simulink Project that can be opened by double-clicking on the project's file "Parking_Meter.prj". This will load and setup the project automatically in MATLAB. Then, in the "PROJECT SHORTCUTS" tab, the list of all the main modules of the project are shown. To open the main model, it is needed to click on "Main Model" in the "GENERAL" section:

Once the model is loaded, it is possible to run the simulation in "Normal" or "Accelerator" modes. It is also possible to build the model as a standalone executable by clicking on the "Build" button:



Note

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`