This chapter describes step by step the build procedure of a Simulink blinky model.

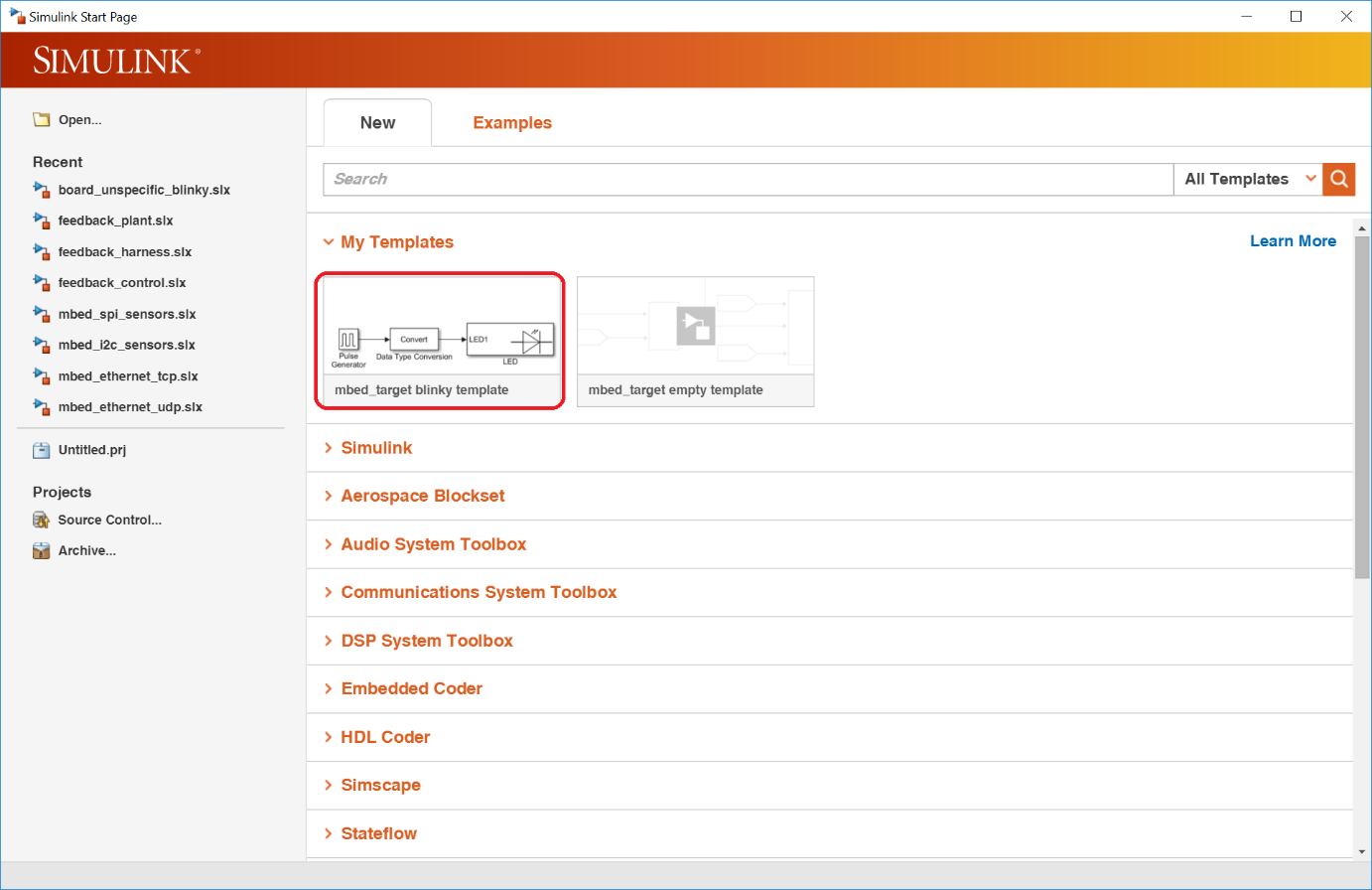
The first variant uses a Simulink template and the second variant build a model from scratch.

Variant 1

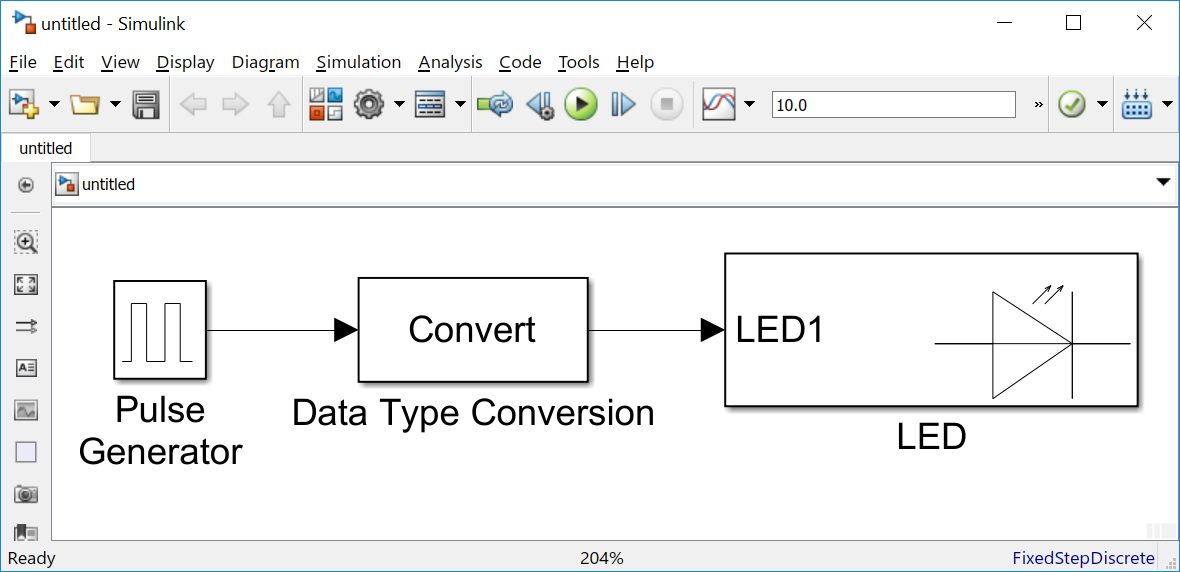
Create a new blinky model with all necessary option already set from the template list:

*>> open\_system(new\_system('untitled','FromTemplate','mbed\_target\_template\_blinky'))*

or out of the Simulink start page:

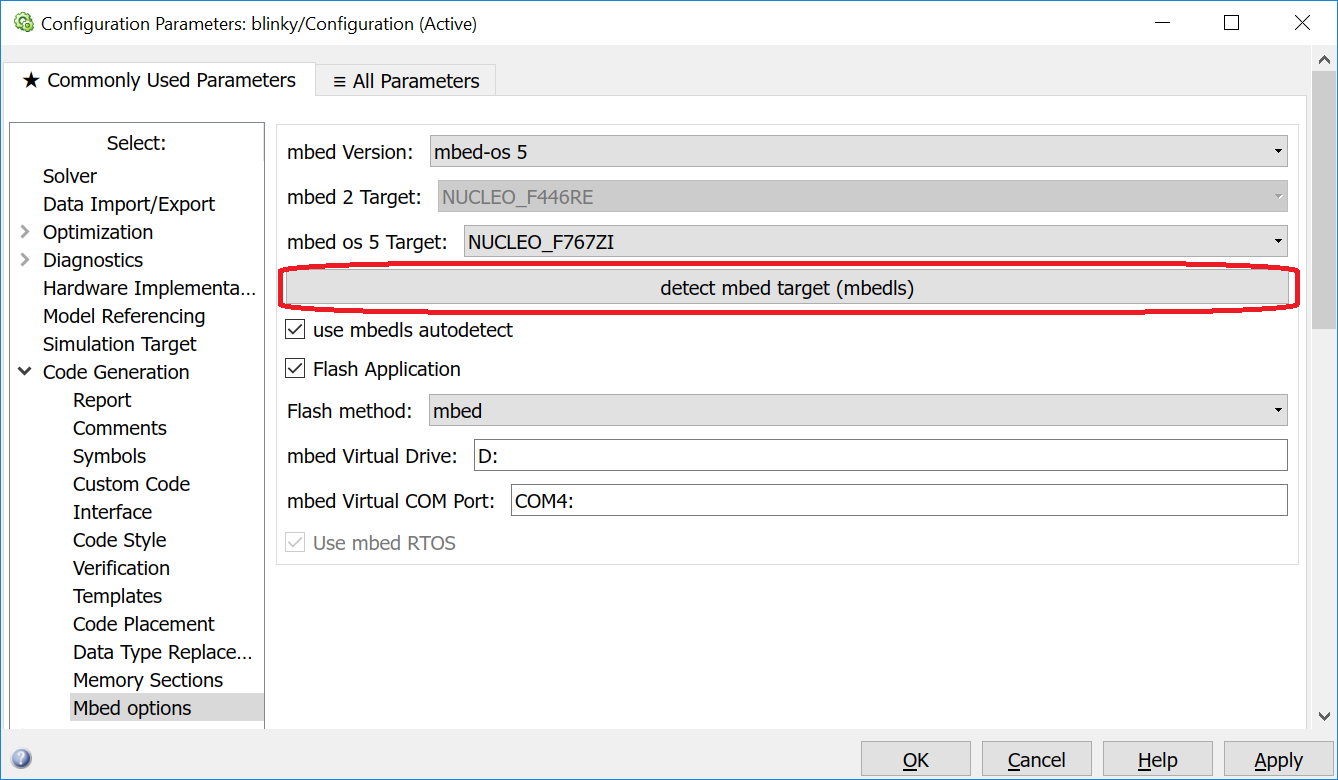


resulting in a new model:



This model **must** be saved before starting the code generation.

The correct mbed target installation can be checked with the integrated target detection. Please open the model configuration and go to **Mbed options**. After pressing the button **detect mbed target (mbedls)** the connected board should be shown in the line above the detect button:



When the connected board does not appear, please check the output of mbedls at the windows commandline. It should look like:

E:\git\mbed\_target\doc>mbedls

+---------------+----------------------+-------------+-------------+--------------------------+-----------------+

| platform\_name | platform\_name\_unique | mount\_point | serial\_port | target\_id | daplink\_version |

+---------------+----------------------+-------------+-------------+--------------------------+-----------------+

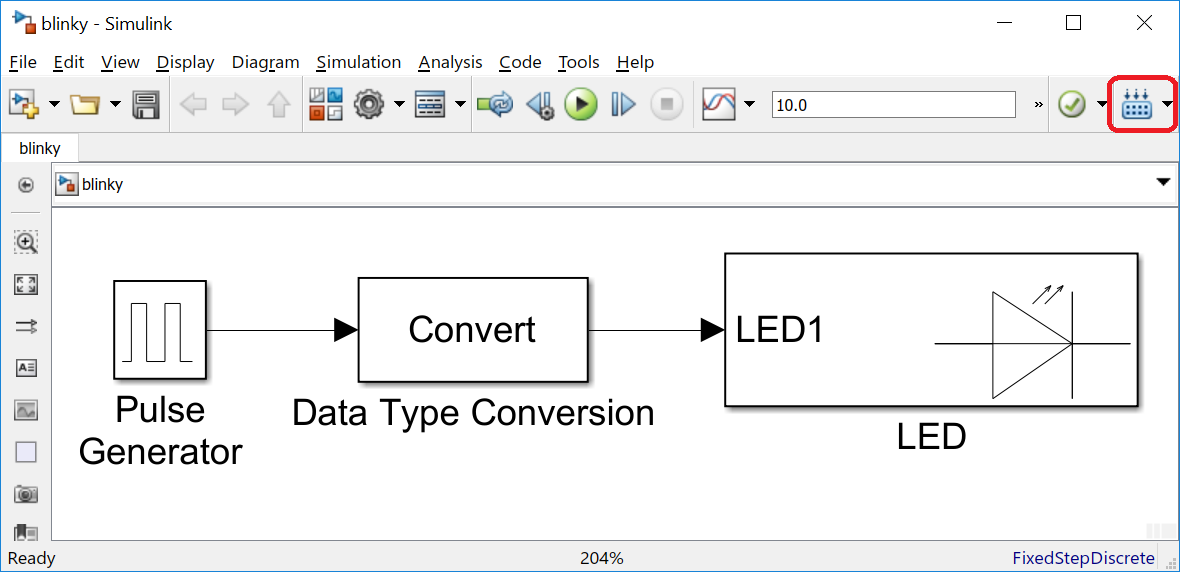
| NUCLEO\_F767ZI | NUCLEO\_F767ZI[0] | D: | COM4 | 08180221053160053A75F920 | 0221 |

+---------------+----------------------+-------------+-------------+--------------------------+-----------------+

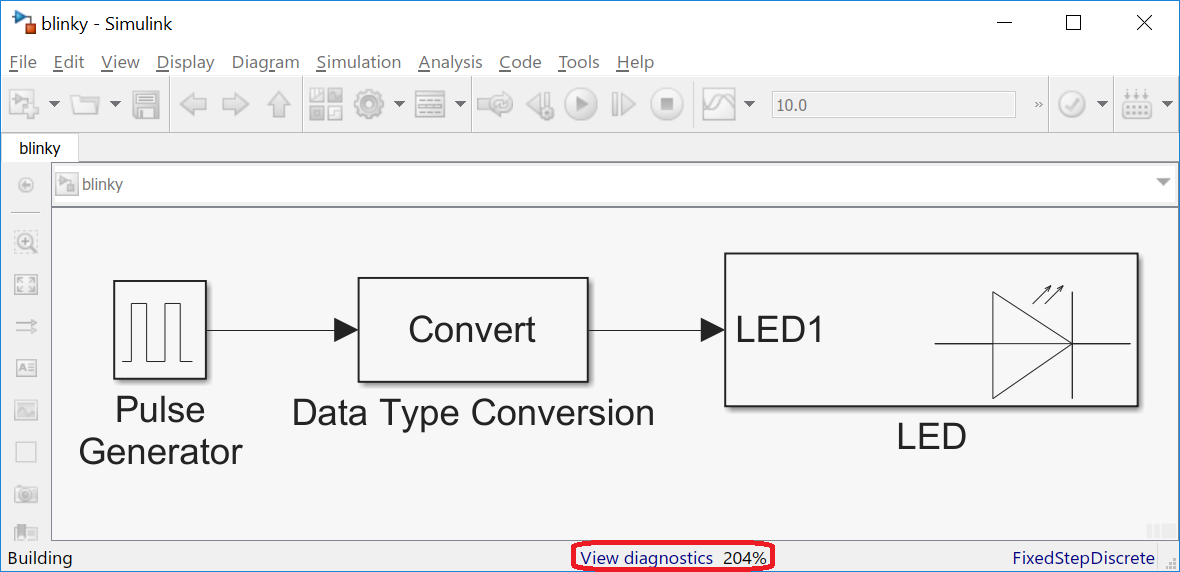
When it does not look like the above example please check your mbed target documentation.

When the board could be detected, the code generation can be started by

* pressing the button **Build model**,
* using the keyboard shortcut **Ctrl+B** or
* choosing the menu item **Code / C/C++ Code / Build Model**.

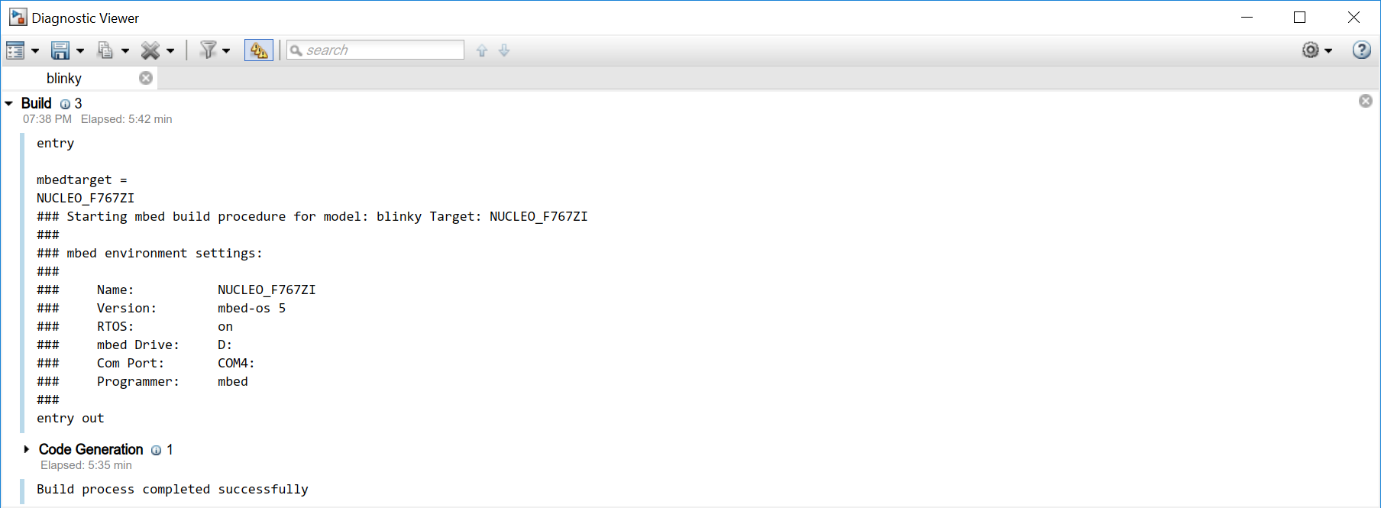


After a short processing time, the diagnostic viewer can be opened:

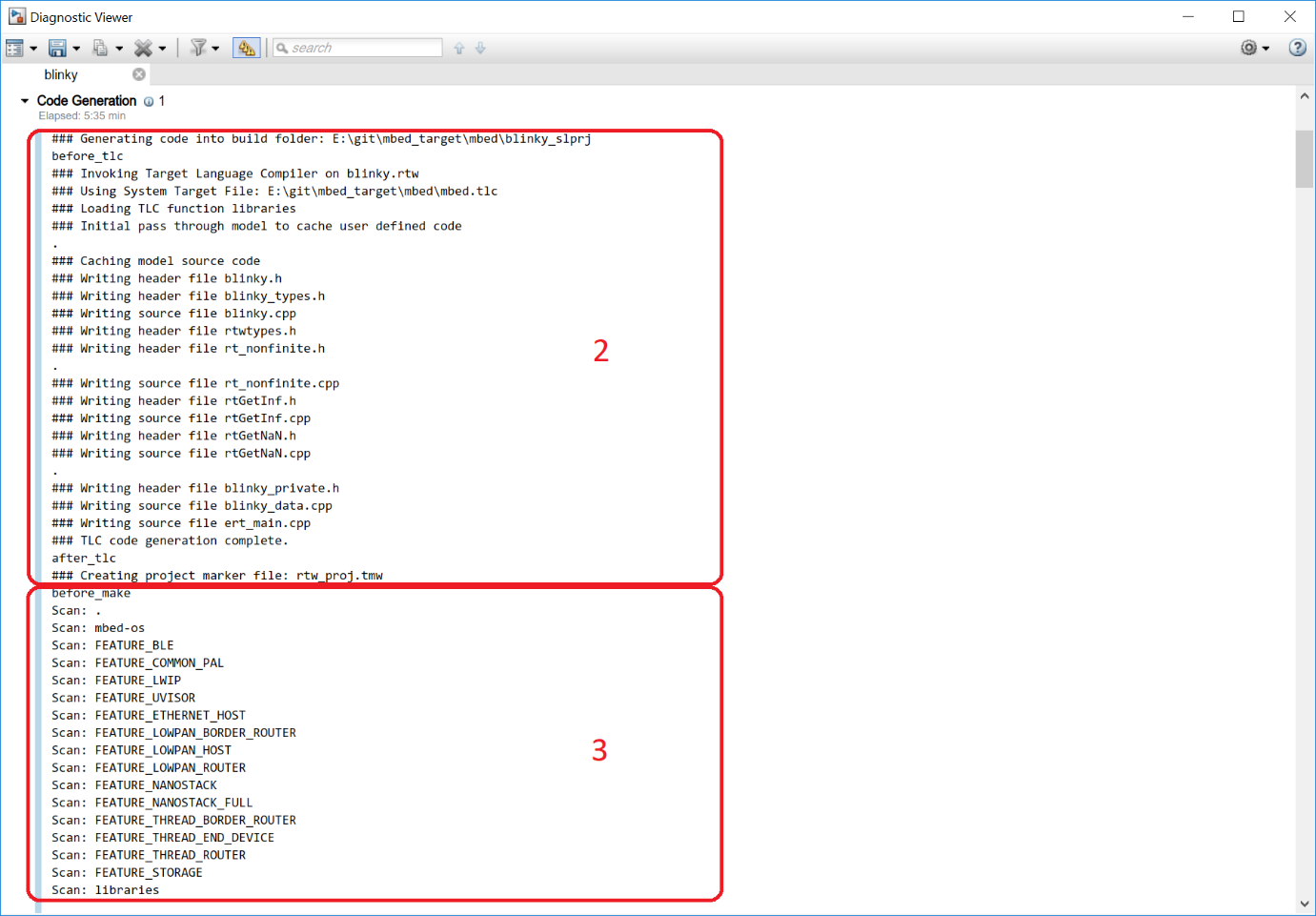


The diagnostic viewer shows the complete code generation process in detail.

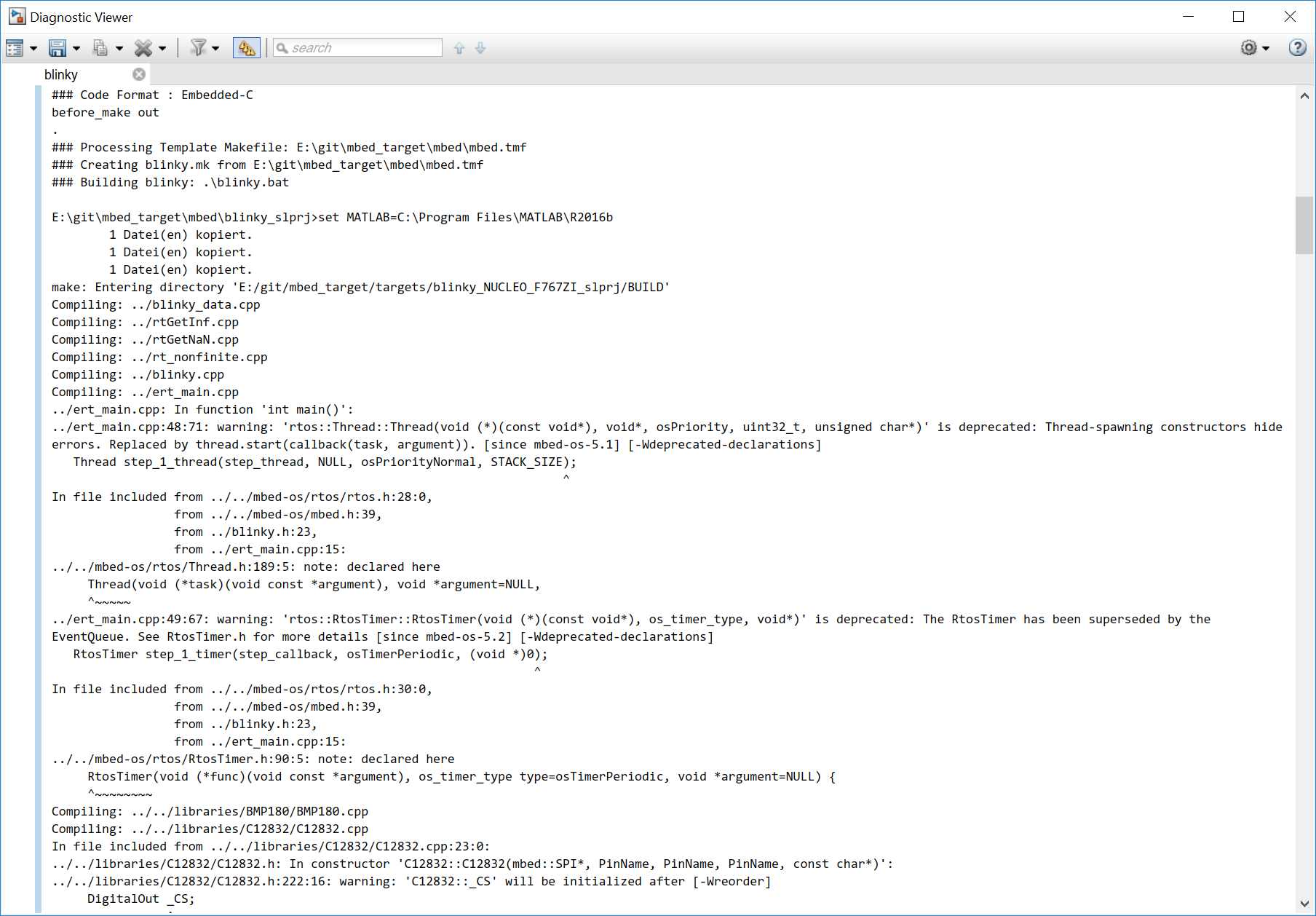
1. The first section shows the mbed target details



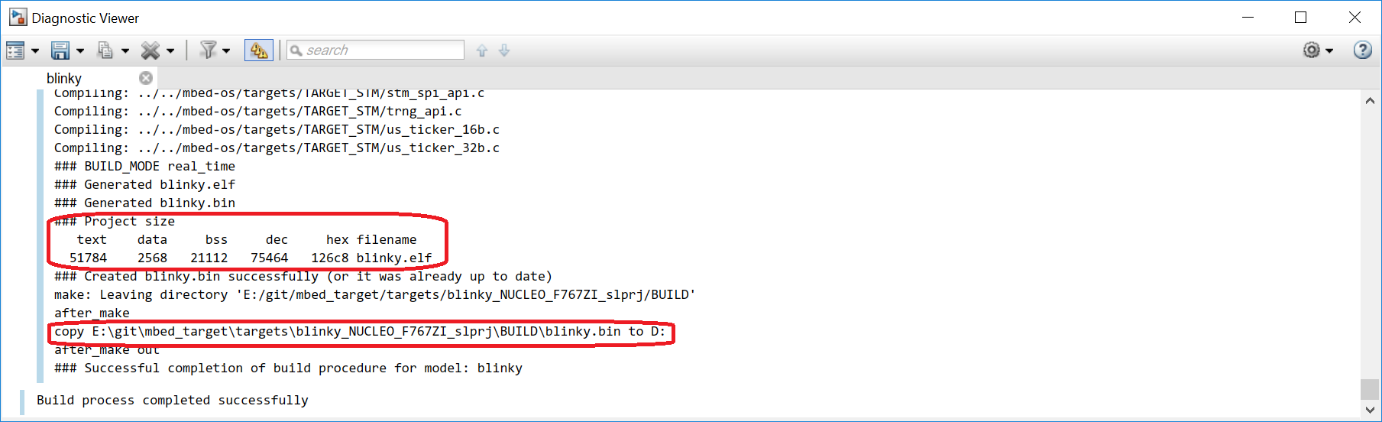
1. The next part contains the MATLAB code generation information
2. The 3rd part is the output of the mbed Python tool generating a Makefile



1. In the next step the mbed library and the generated C++ code is compiled. When the Simulink model is build the first time, this process can take a while.

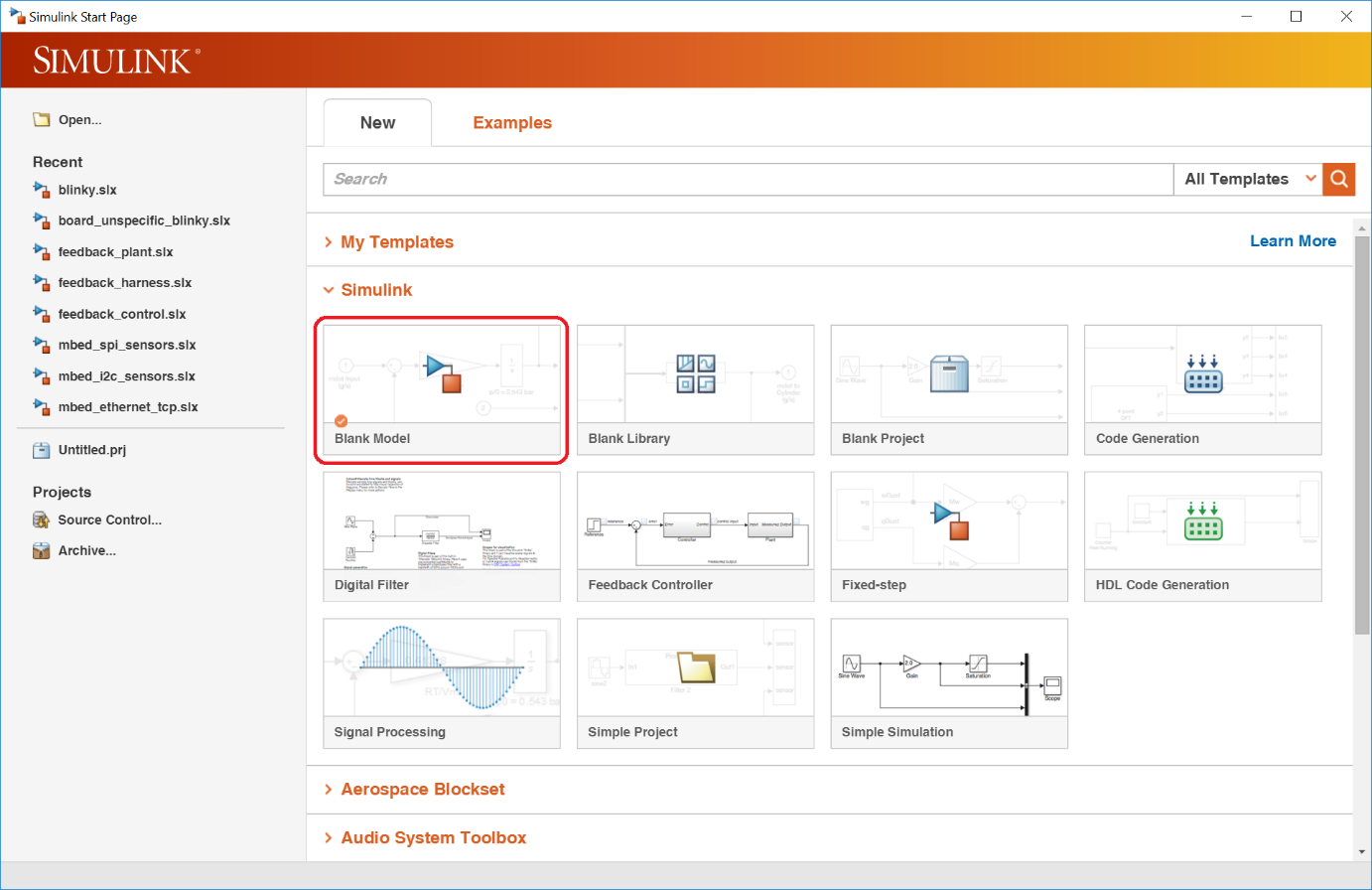


1. Finally, after a successful build, the sizes of the different sections (e.g. text – instruction code, data+bss – data memory) should be displayed. In one of the last lines the result of the flashing can be seen. For an mbed board it is just a copy command.

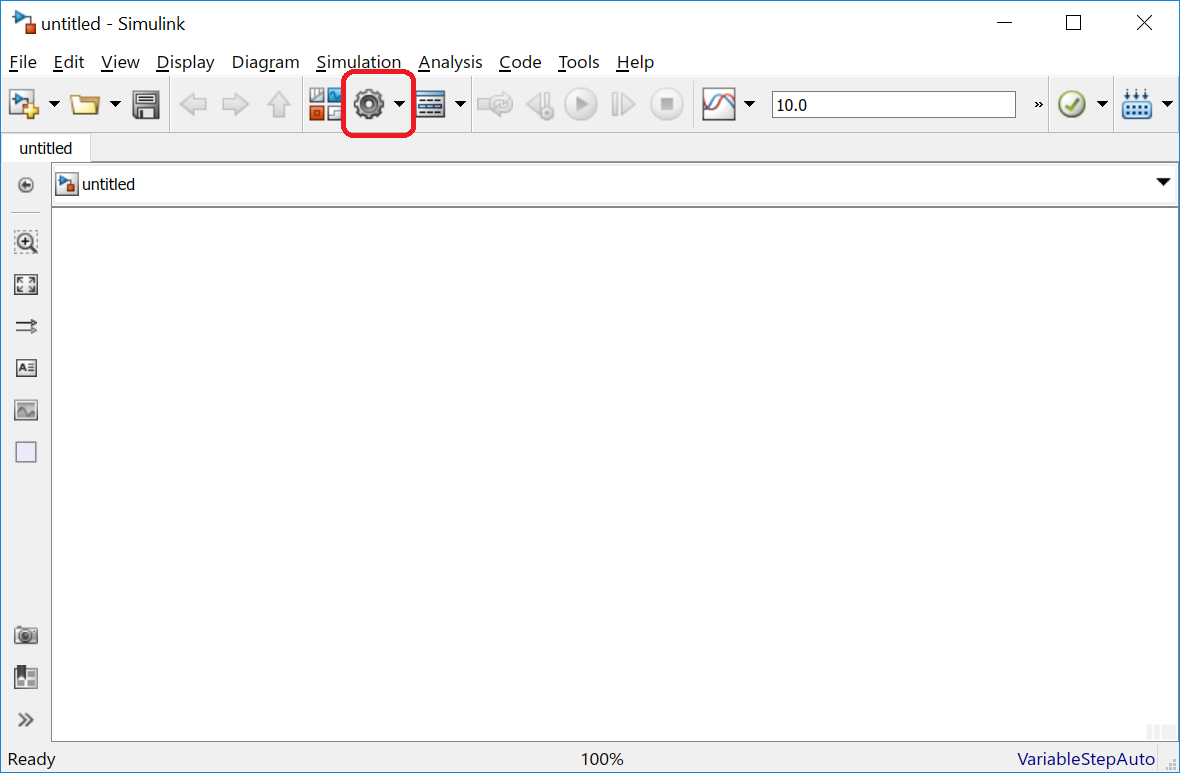


Variant 2

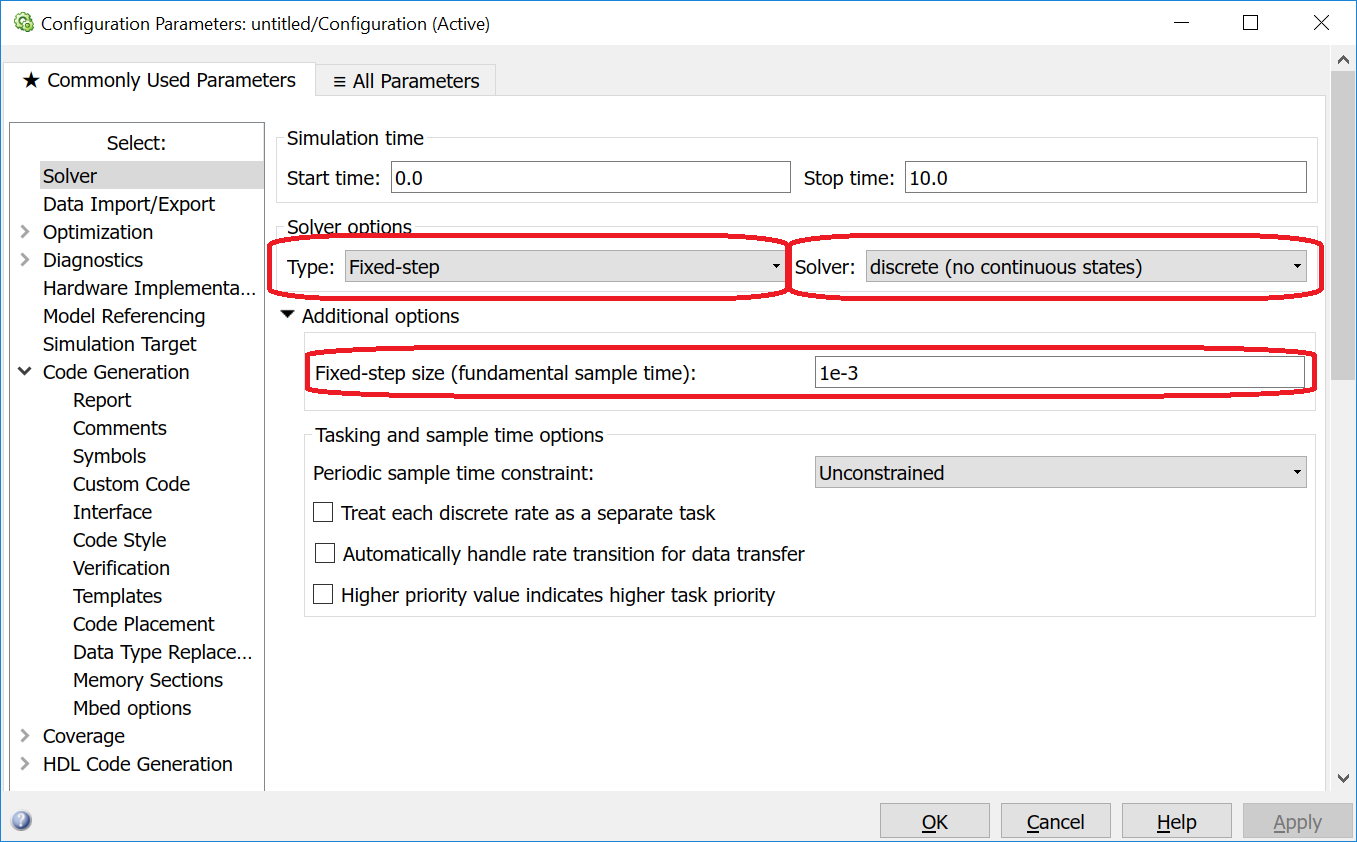
Create an empty Simulink model:



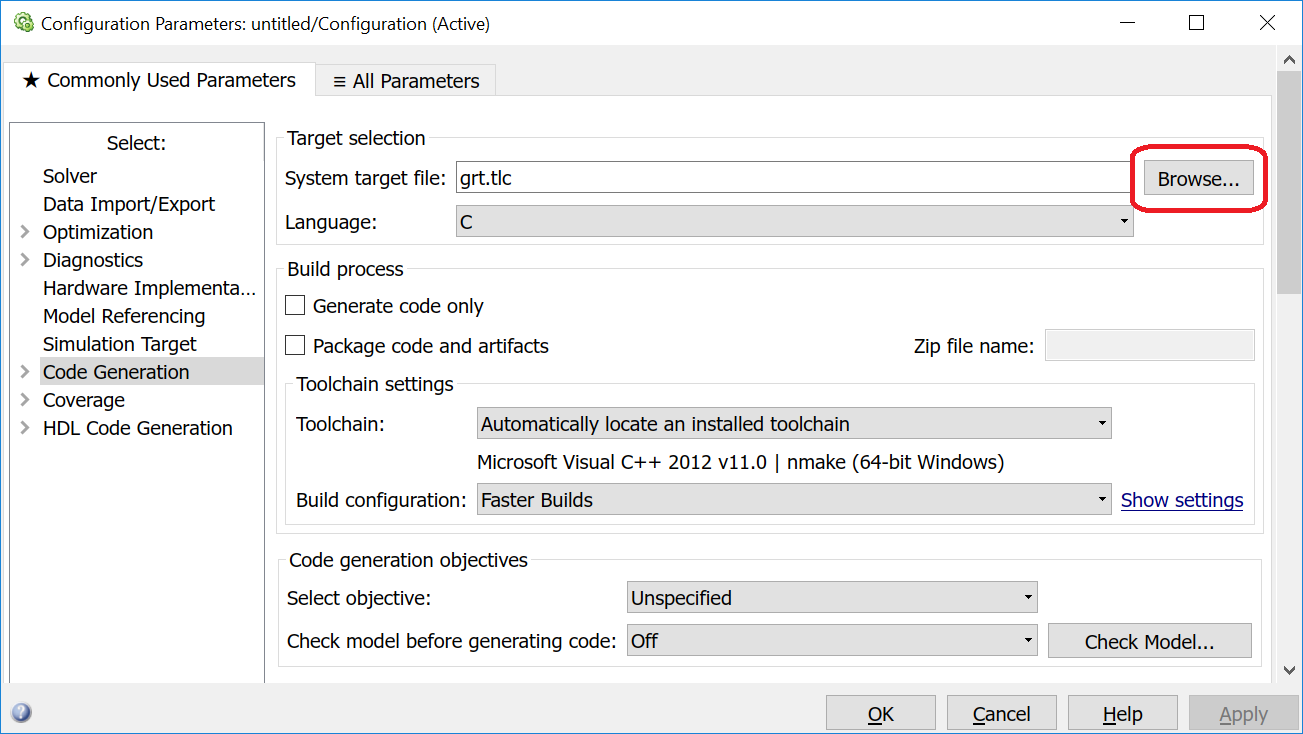
Open the model properties:

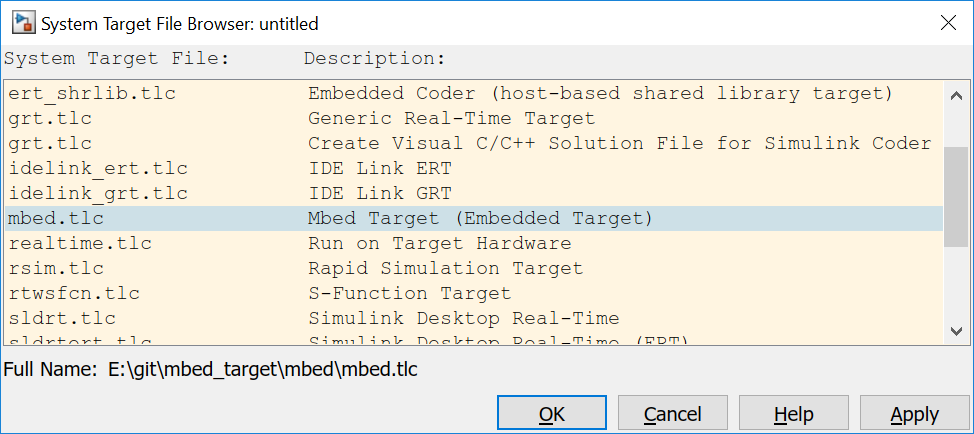


As solver options **Fixed-step** and **discrete** must to be chosen. The **fixed-step size** can be configured to 1ms. Other values are possible. The minimal value depends on the mbed target capabilities.

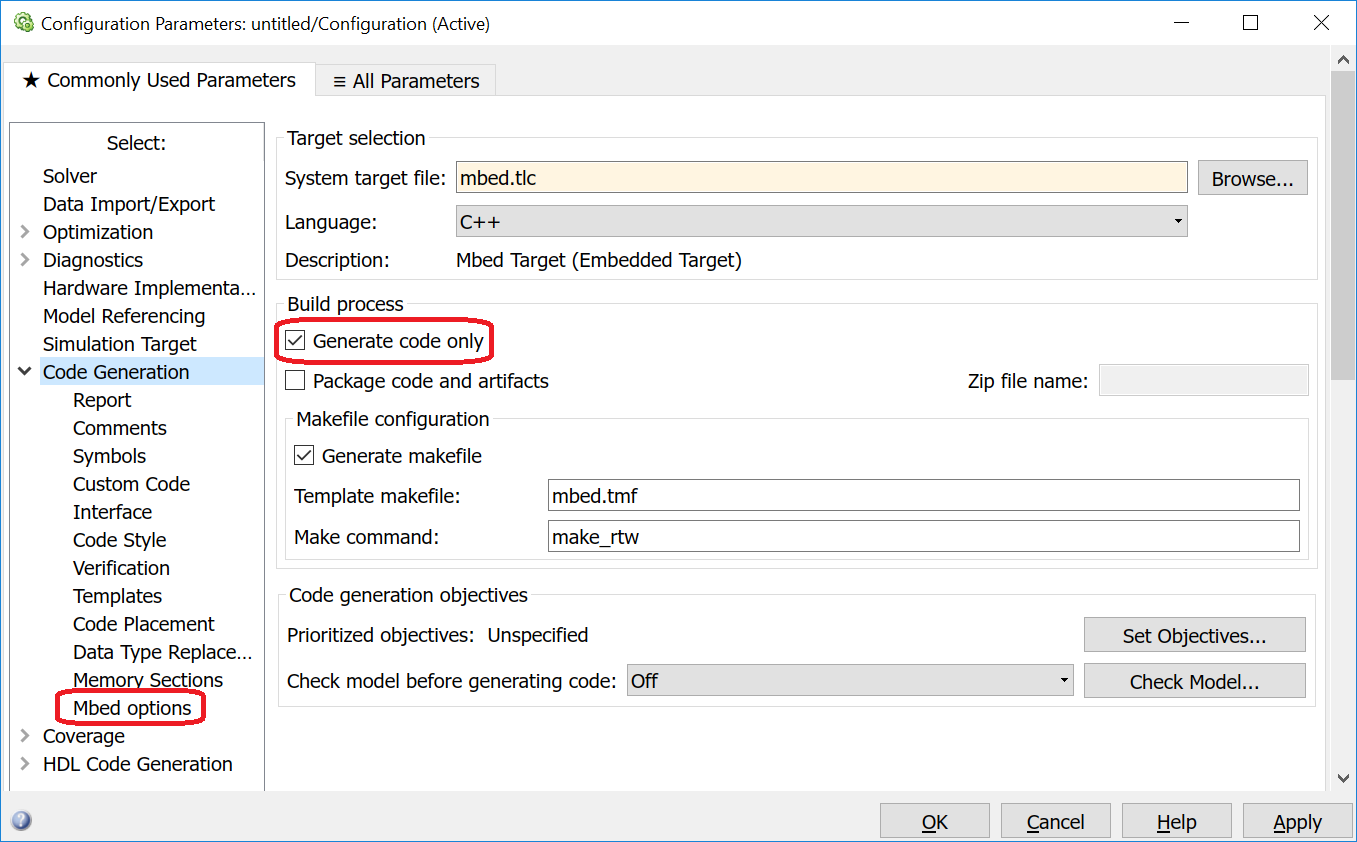


Under **Code Generation** the mbed.tlc has to be chosen:

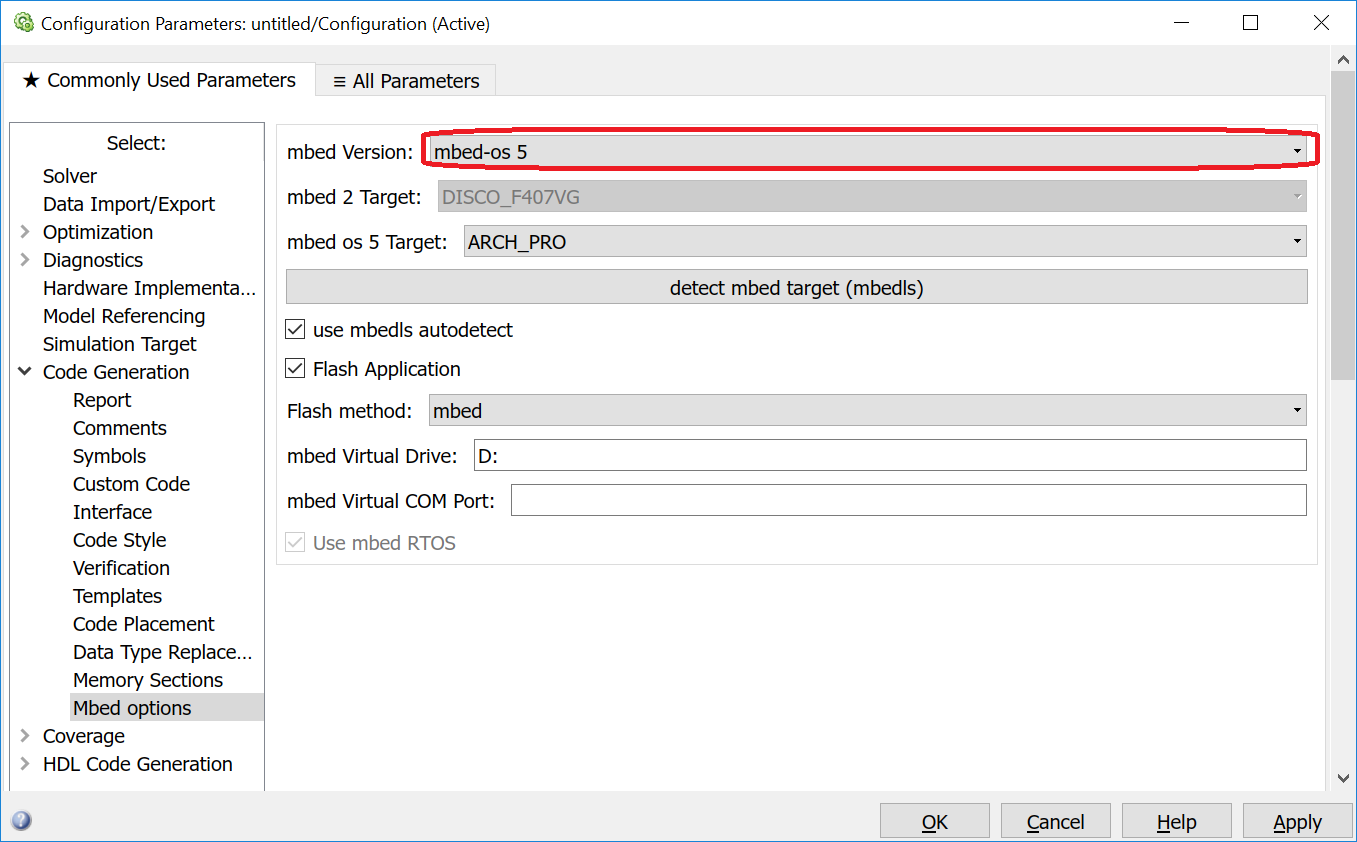




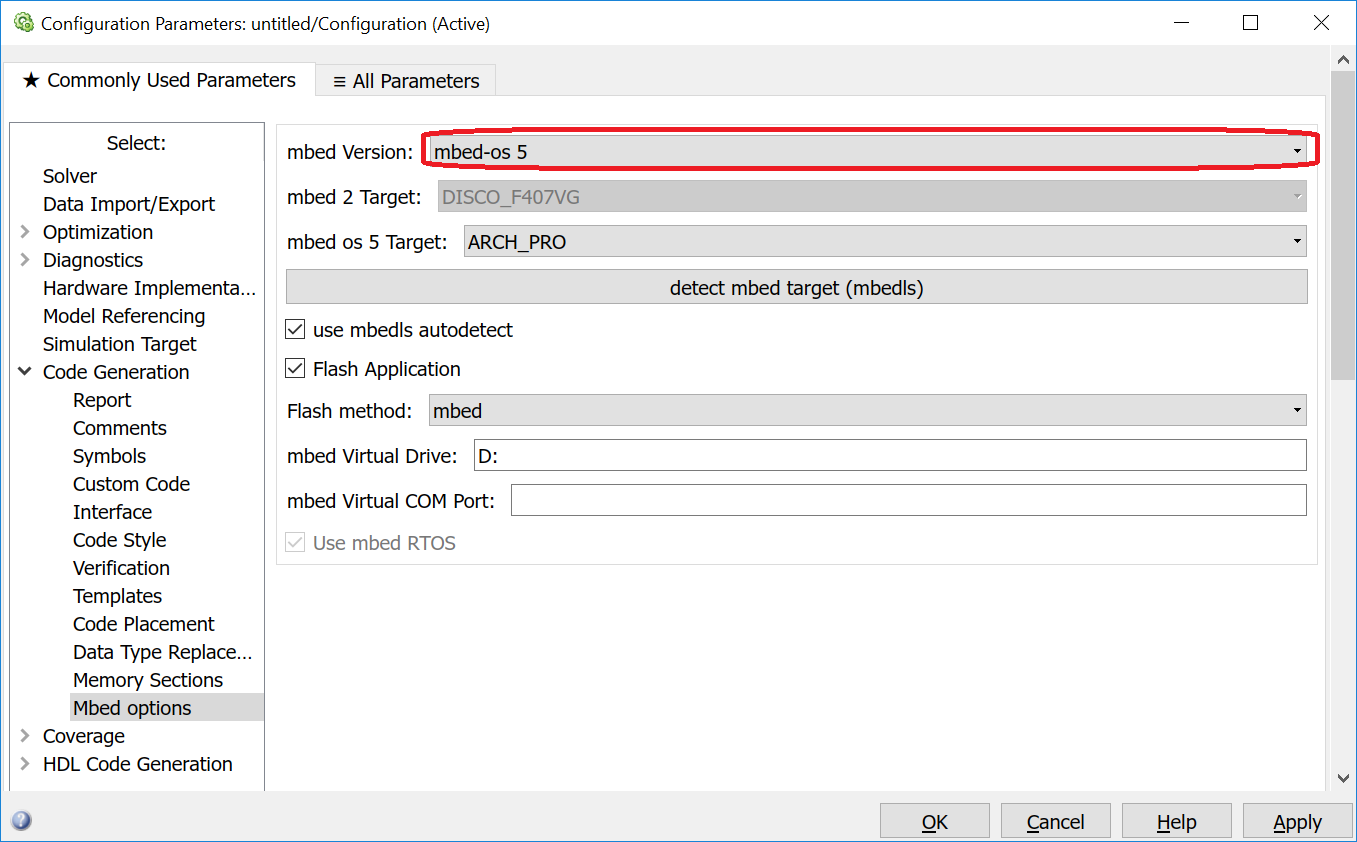
To compile the generated C++ code, the check box **Generate code only** has to be switched of. Under the menu **Mbed options** the mbed\_target specific options are available.



Please chose mbed os 5:

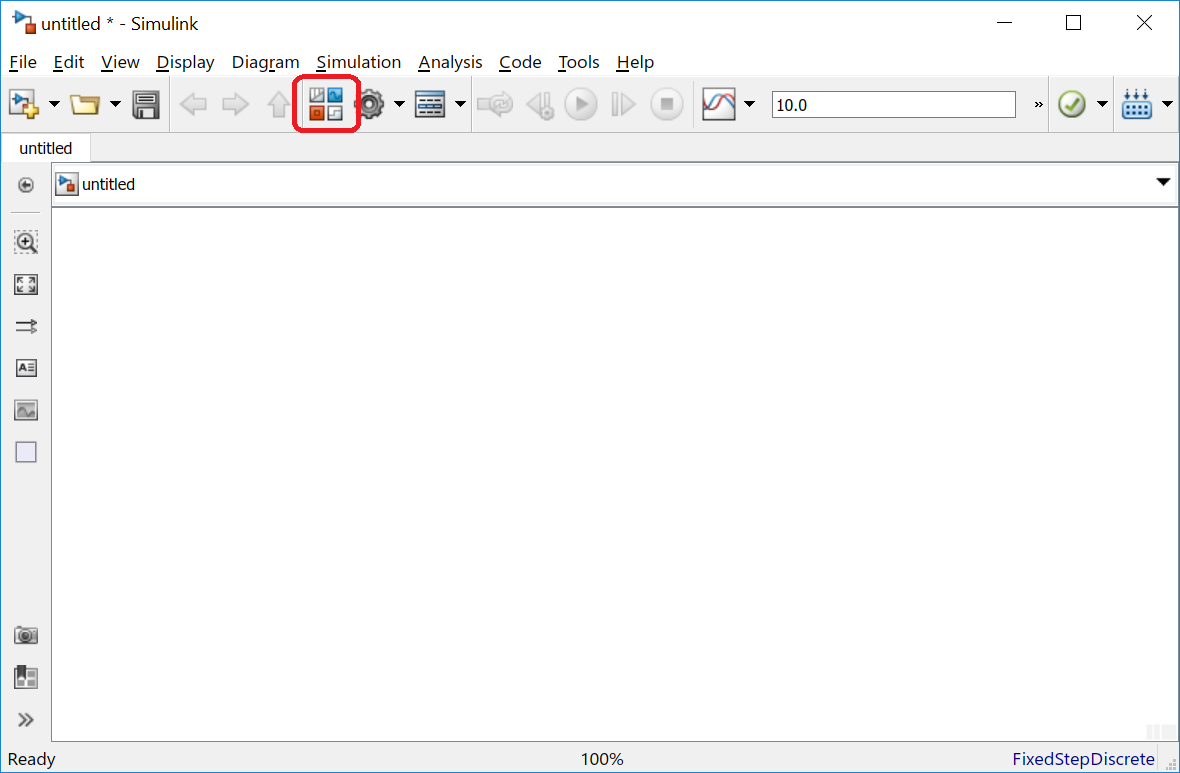


and check if the mbed board is correctly connected:



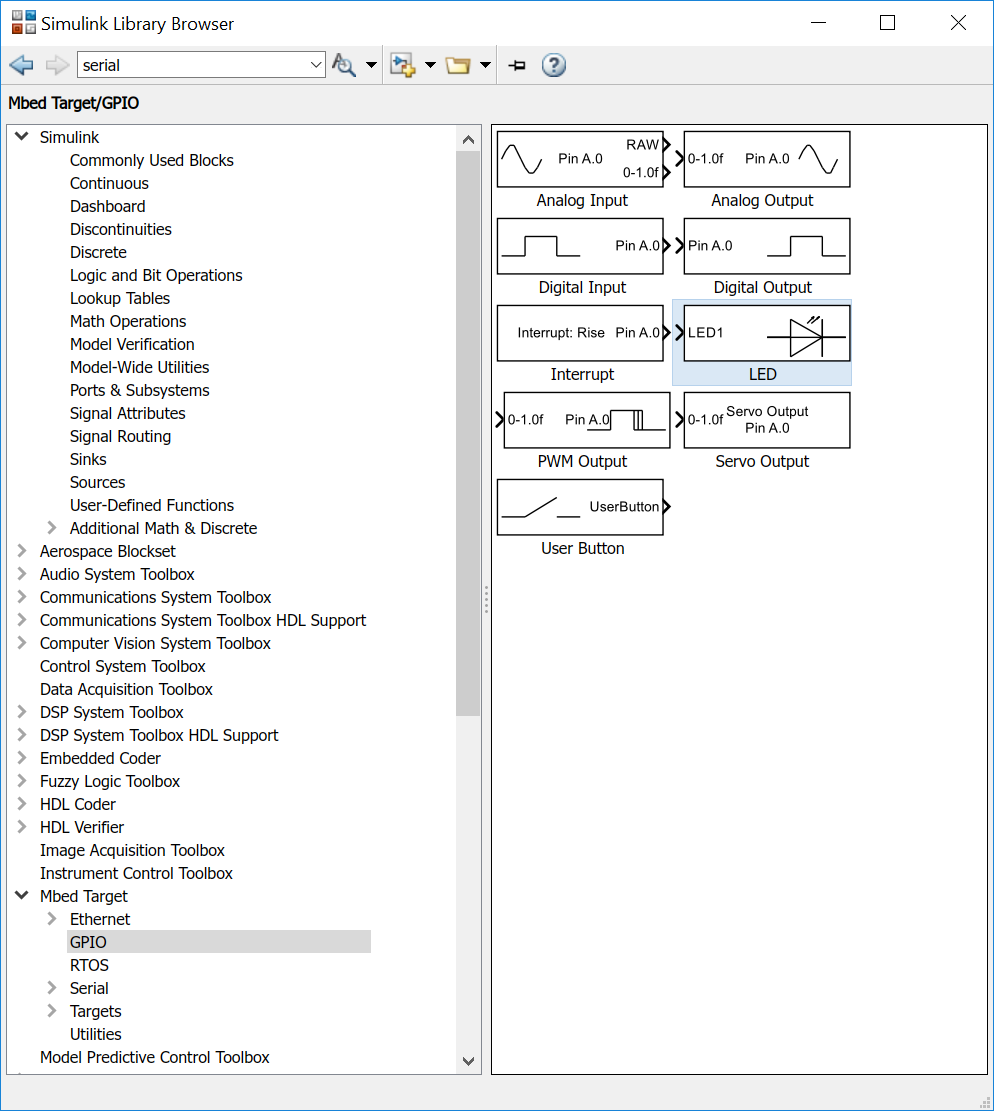
The result is identically to the step in the 1st Variant.

To build the model, one possibility is the usage of the library browser:



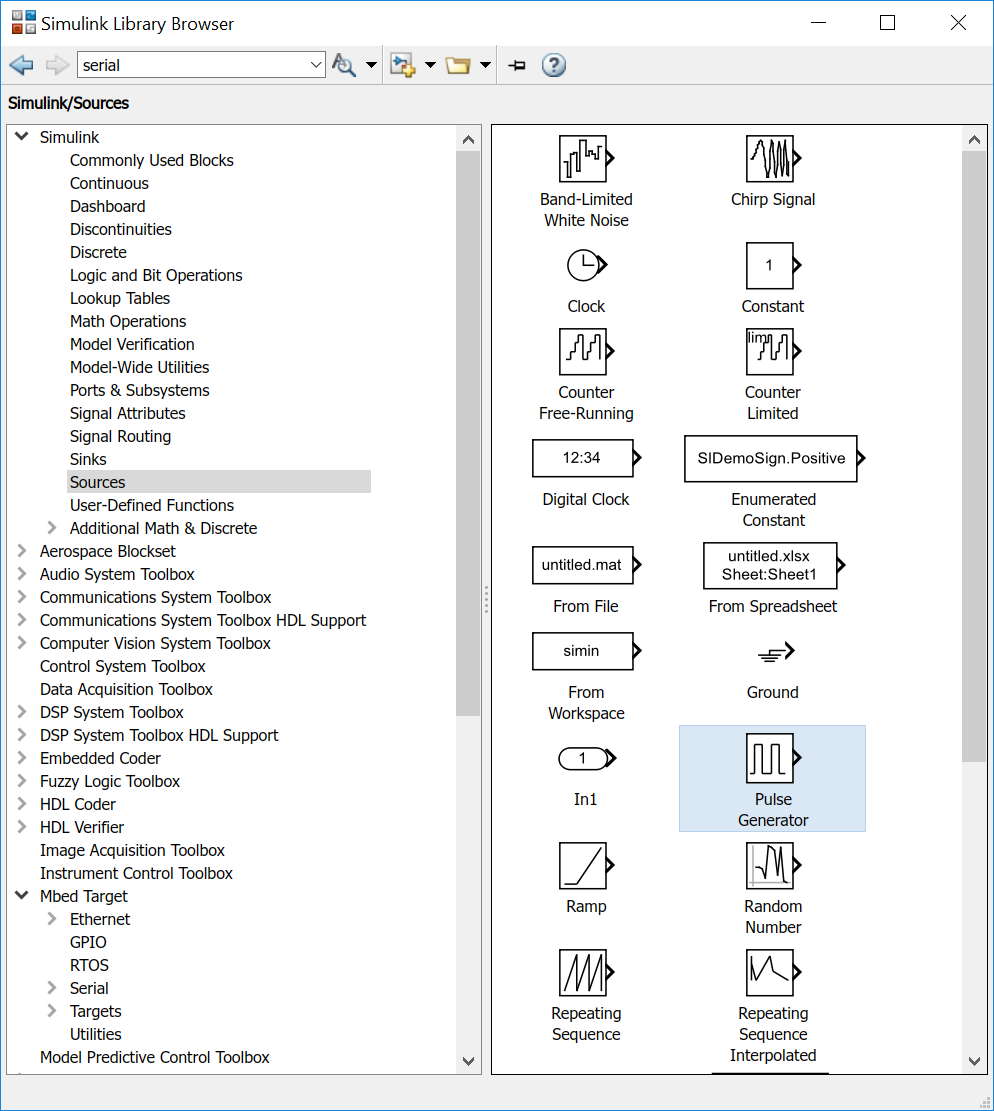
Simulink allows the usage of a huge amount of blocks for the code generation. But not every block can be used.

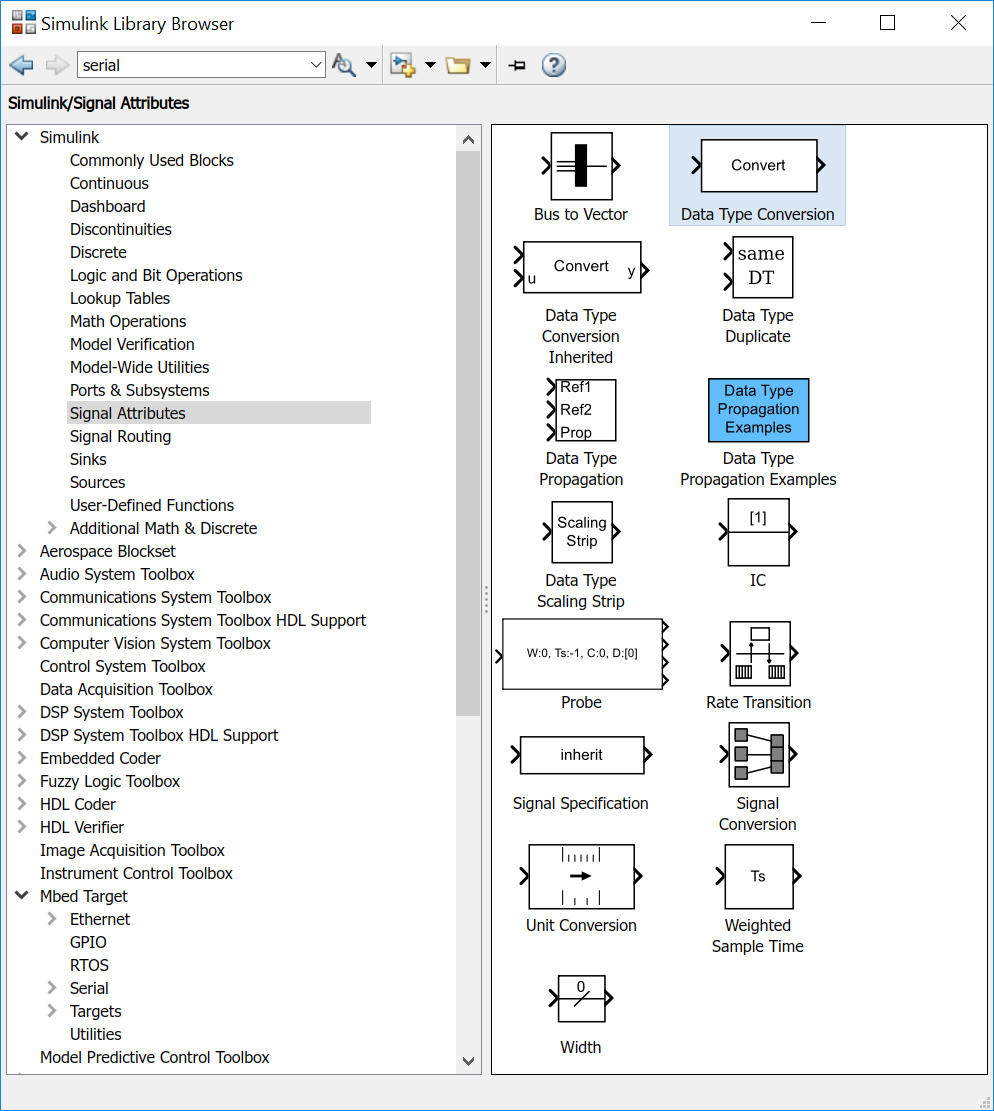
mbed\_target delivers an own subtree in the Simulink library browser with the target specific blocks:



These blocks are described in this documentation. Alternatively the Simulink block help function can be used.

Please add the marked LED, Pulse Generator (with amplitude 1) and Data Type Conversion blocks:





After connecting the three blocks the model is identic to the model under Variant 1. The code generation can be executed as already described there.