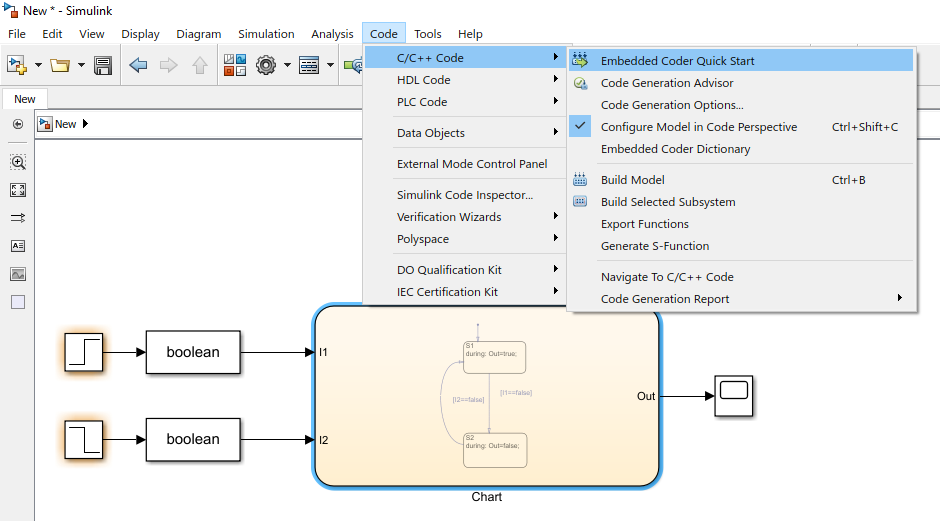
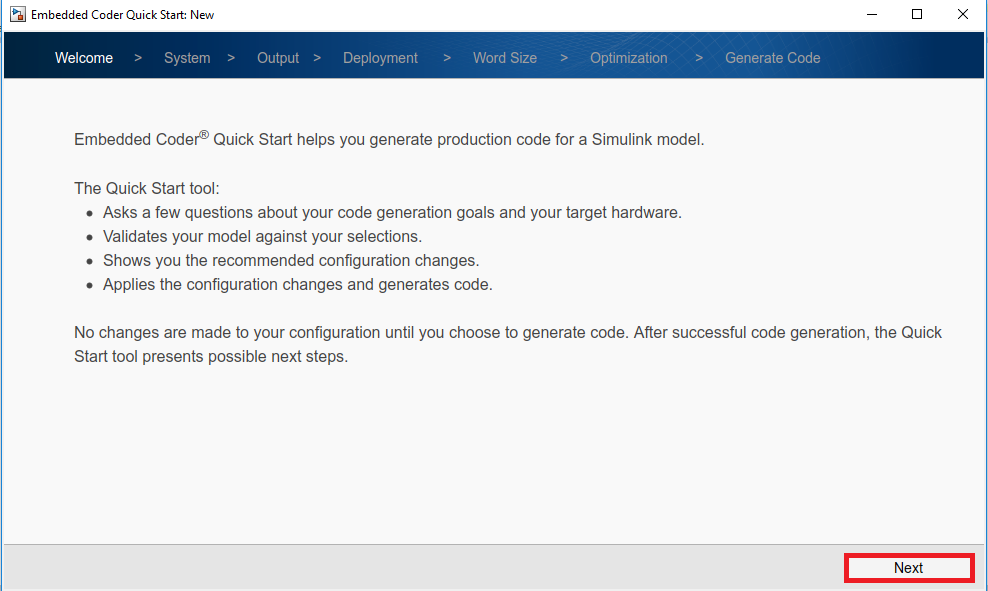
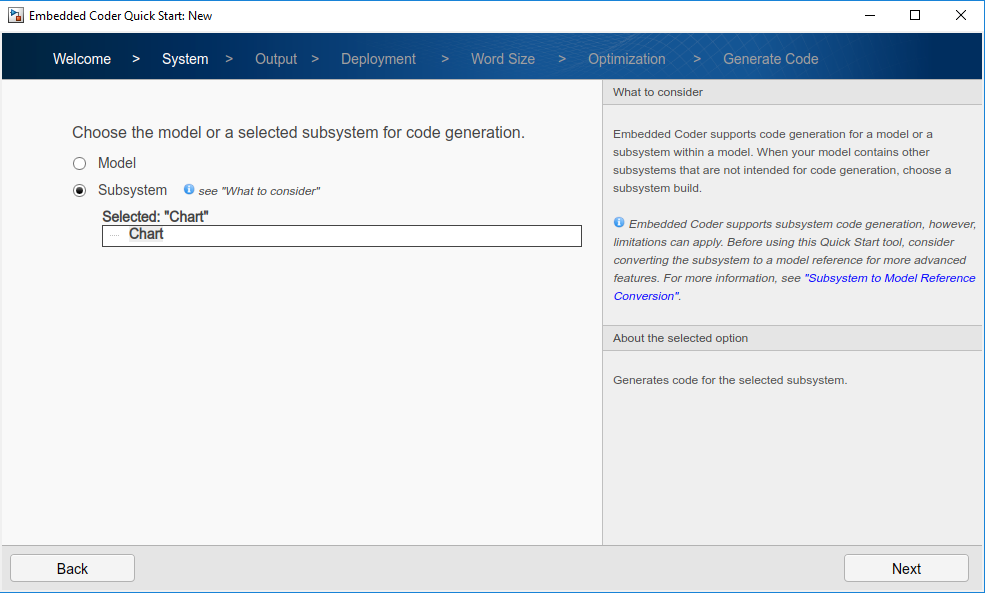
To compile a stateflow chart, you need the Matlab Compiler. Call the compiler in Simulink via Code -> C/C++ Code -> Embedded Coder Quick Start.



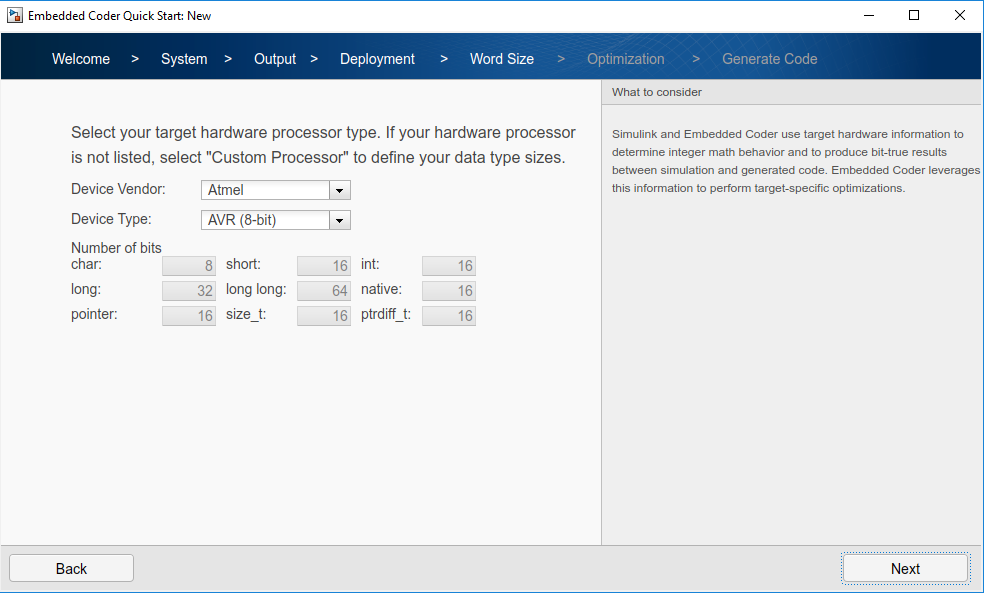
Go through the wizard.



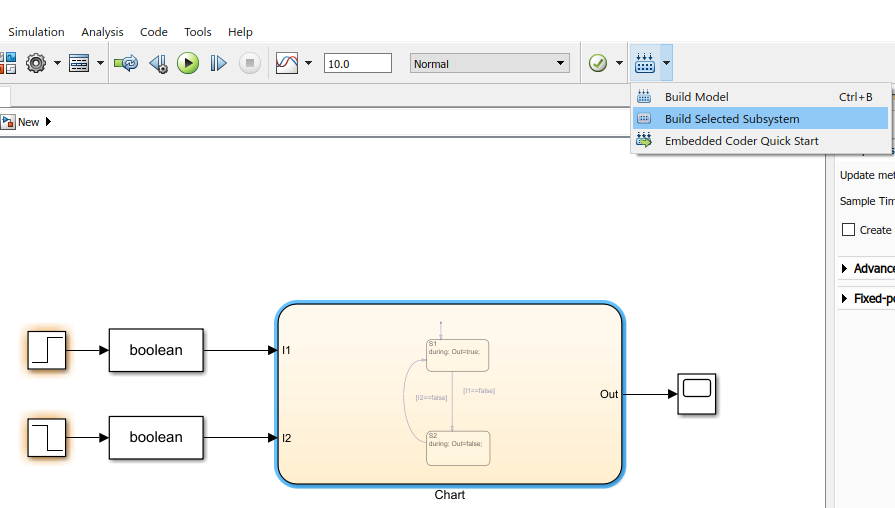
Select the subsystem, which you want to compile.



Select the target processor.



After the wizard build the selected subsystem.



Copy from folder \*Name\_Submodule\*\_ert\_rtw the following files to your Arduino project:

* \*Name\_Submodule\*.h
* \*Name\_Submodule\*.cpp
* rtwtypes.h

Switch to Arduino IDE and include both header files:

#include "Chart.h" // Model's header file

#include "rtwtypes.h" // Type definitions

Initialize chart object:

**static** ChartModelClass rtObj;

Call in the setup function the initialize function of the chart object:

rtObj.initialize();

In the loop function check first, any stateflow error and perform the error proccdure:

**if** rtmGetErrorStatus(rtObj.getRTM()) != (**NULL**) {

// Called when error in stateflow

Serial.print("StateFlow Error!");

**while** (true) {}

}

Otherwise write the input values to the stateflow, perform a stateflow calculation and write the output values to the dedicated outputs:

**else** {

// Called when no error in stateflow

// Write inputs to stateflow

rtObj.rtU.I1 = digitalRead(BTN1);

rtObj.rtU.I2 = digitalRead(BTN2);

// Call stateflow

rt\_OneStep();

// Write output value to output pins

digitalWrite(LED, rtObj.rtY.Out);

}

In the \*Name\_Submodule\*.h comment the following lines:

#include "rtw\_continuous.h"

#include "rtw\_solver.h"