FAKULTA
ELEKTROTECHNICKÁ
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Základy generování aplikačního SW ze Simulinku

Mikroprocesorové řízení pohonů 2

Jakub Talla

www.rice.zcu.cz

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj pro inovace

OSNOVA PŘEDNÁŠKY č. 9

1) Úvod do generování kódů ze Simulinku

2) Automatické generování aplikačního SW ze Simuinku

3) Testování vygenerovaného SW pomocí SIL/PIL

Úvod do generování kódů ze Simulinku

Motivace automatického generování C kódu ze Simulinku

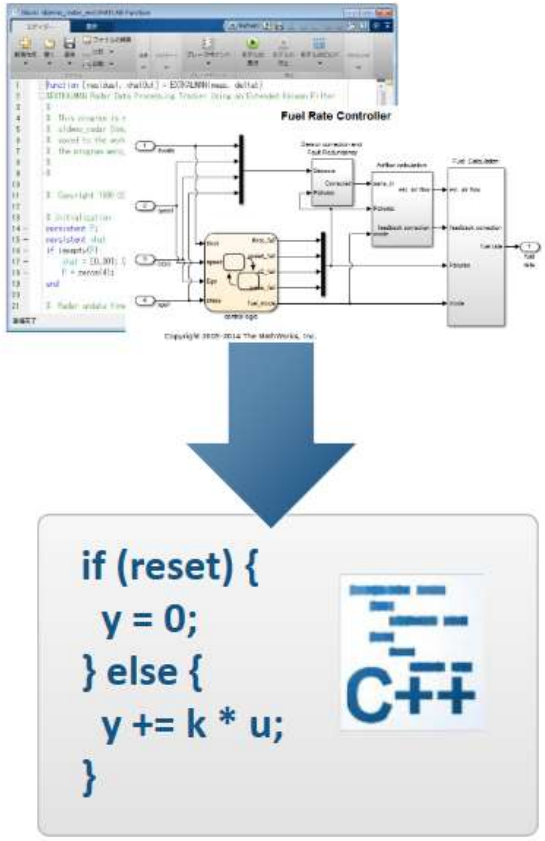
Modelování a Simulace

- **Grafické modelování**

  čitelnost, srozumitelnost, přenositelnost

- **Brzká verifikace a validace**

  (průběžná validace modelu, kódu SIL… HIL apod.)



Automatické generování kódů

- **Redukce programátorského času**
- **Redukce chyb při transformaci algoritmu do kódu**
- **Propojení modelu a kódu**
- **Automatizace testů**

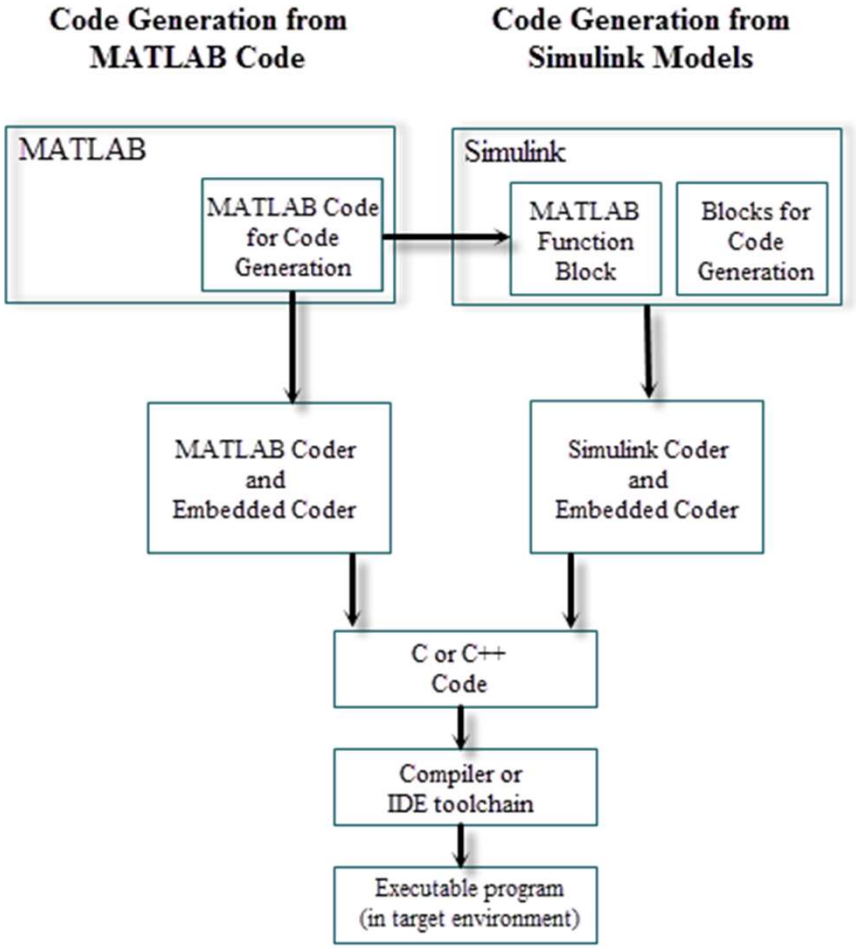
Úvod do generování kódů ze Simulinku

Příklad automatického generování na SW ECU

	Hand Code	Auto-code	Percent Change
Calibration ROM	9464	9464	0%
Code ROM	2952	2900	-1.76%
RAM	240	238	-0.83%

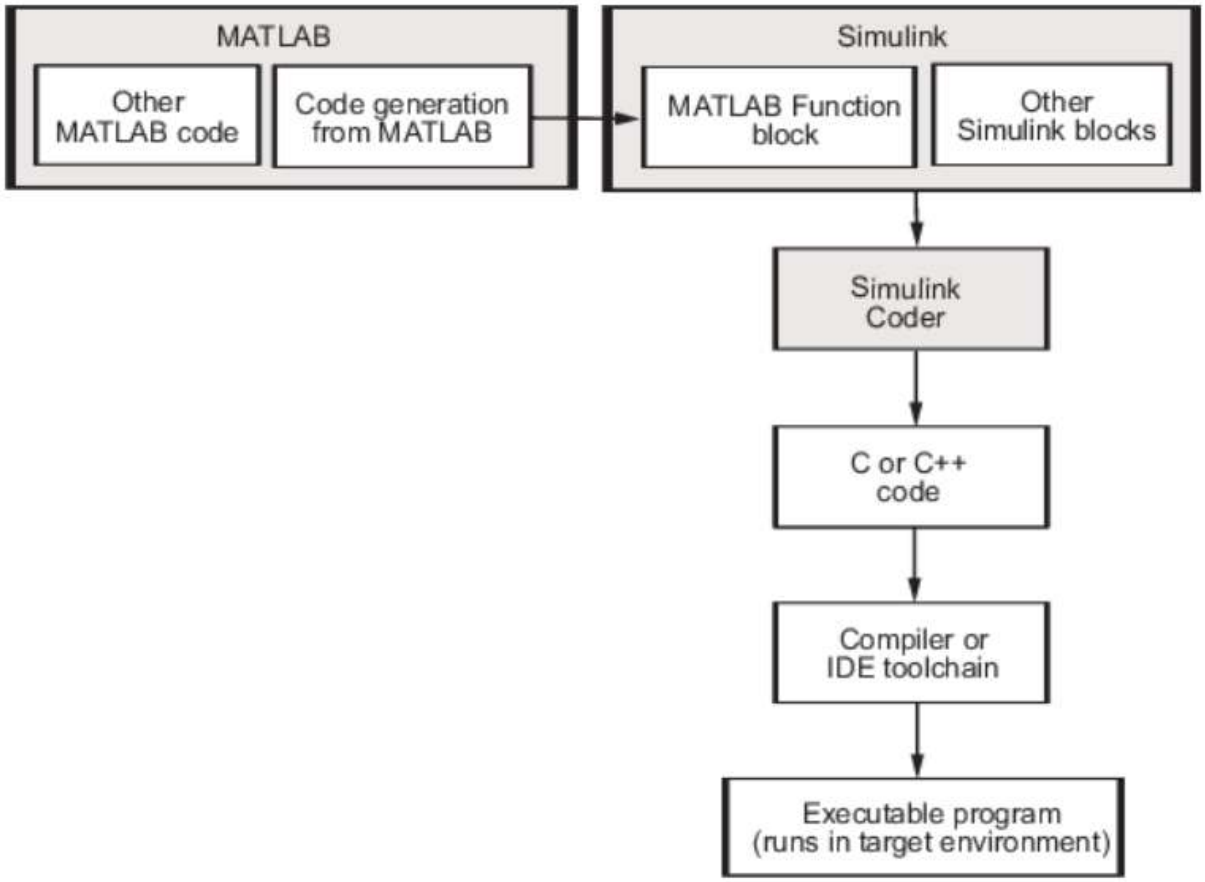
More Efficient Than Hand

Úvod do generování kódů ze Simulinku



Úvod do generování kódů ze Simulinku

Automatické generování C kódu ze Simulinku



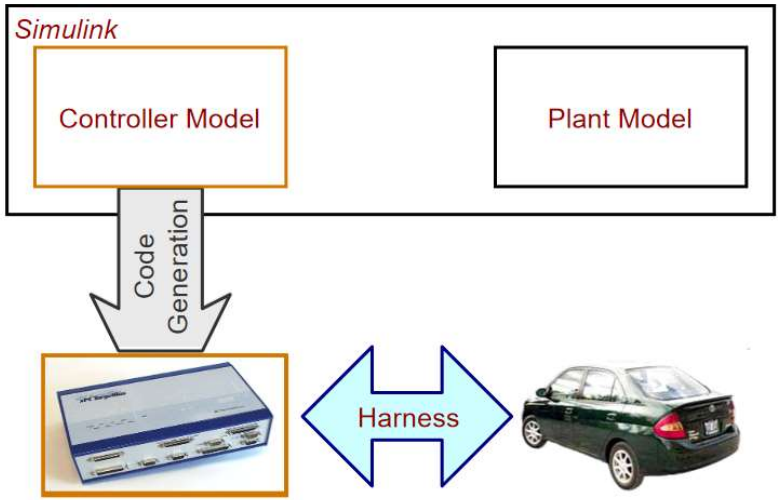
Úvod do generování kódů ze Simulinku

Pro jaké účely se hodí generování kódu ze Simulinku a Stateflow?

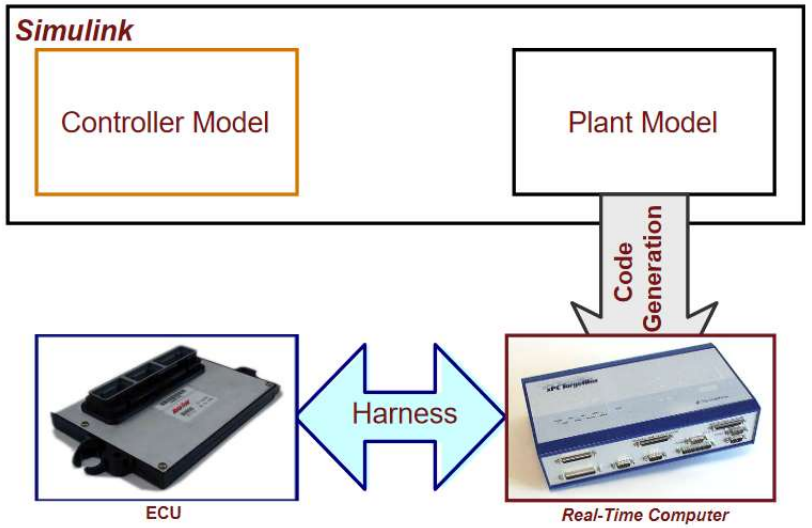
- Akcelerace Simulací (mód rapid accelerator)
- Rapid prototyping (prototypování aplikací)
- Hardware In the Loop (HIL) testování
- Vývoj aplikačního embedded SW

Úvod do generování kódů ze Simulinku

Rapid Prototyping



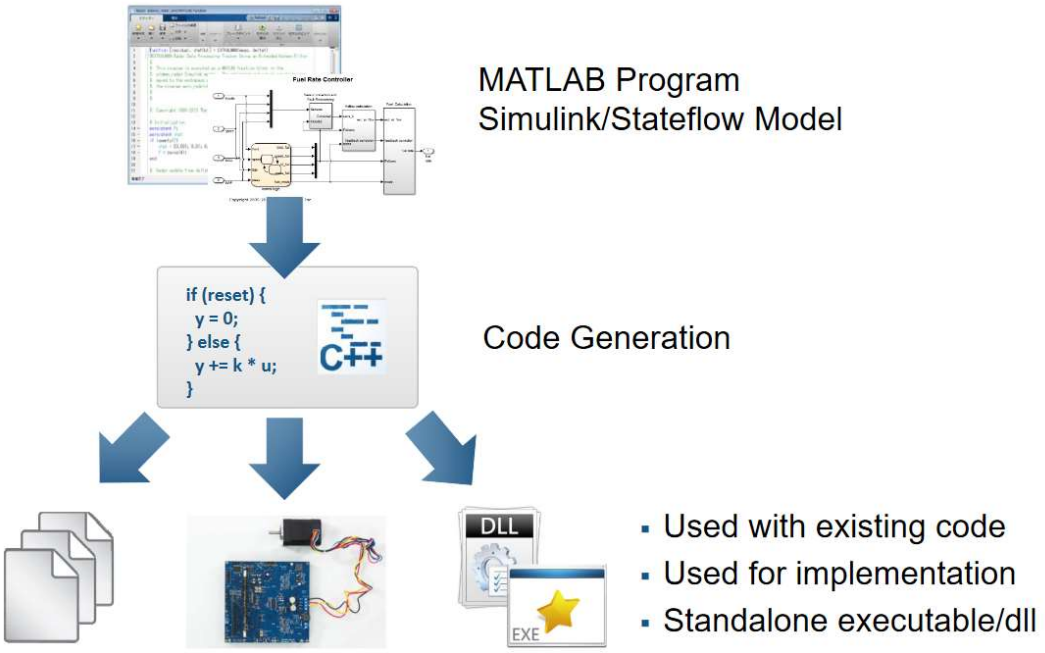
Hardware in Loop (HIL)



Úvod do generování kódů ze Simulinku

Kód může být vygenerován v různých podobách:

1) Aplikační SW takřka nezávislý na HW - nutná vlastní integrace do projektu
2) Projekt s nastavením základních HW komponent (interrupty apod.), další HW funkce se ručně importují do vygenerovaného projektu
3) Kompletní projekt (není nutné SW ručně upravovat)

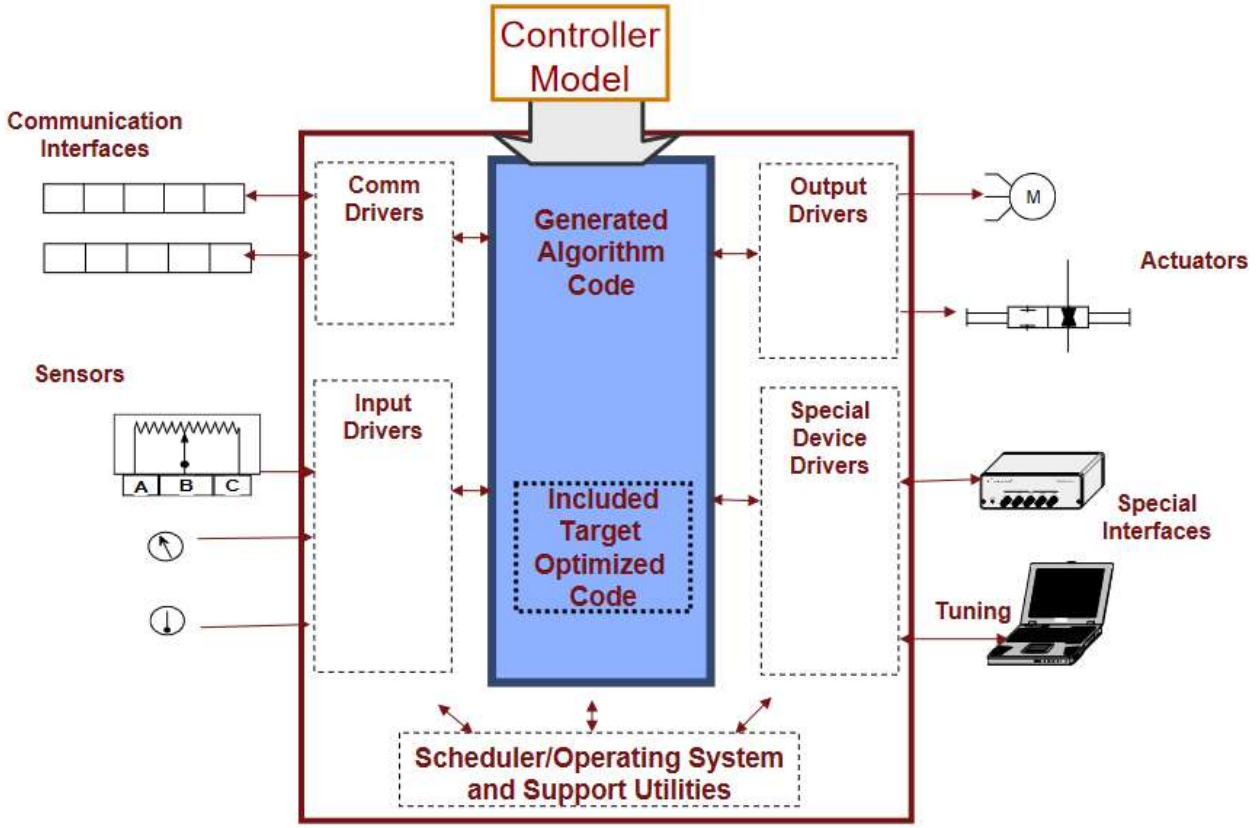


Vývoj a integrace aplikačního SW pomocí Simulinku

Generování aplikačního SW ze Simulinku

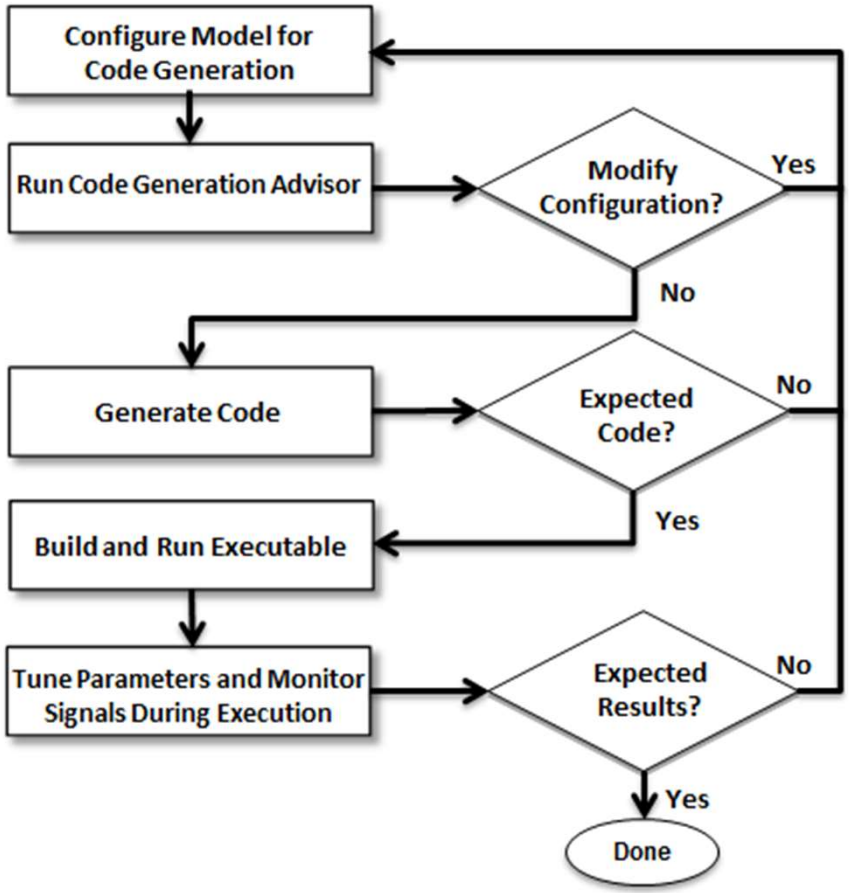
Generujeme pouze funkce nezávislé na HW a provádíme jejich otestování a integraci v IDE (např. CCS)

Tj. dnes se nebudeme bavit o nastavení HW, interruptů a schedulingu



Generování aplikačního SW ze Simulinku

Průběh generování kódu ze Simulinku



Generování aplikačního SW ze Simulinku

Postup při návrhu:

1) Úprava modelu pro generování

přechod ze spojité oblasti do diskrétní, double -> fixed point, optimalizace, pojmenování signálů apod.

2) Konfigurace modelu pro generování kódu

Konfigurece v model settings – nastavení záložek Solver, HW Implementation, Code Generation

3) Výběr co chci generovat

Model reference, subsystem, function

4) Otestování

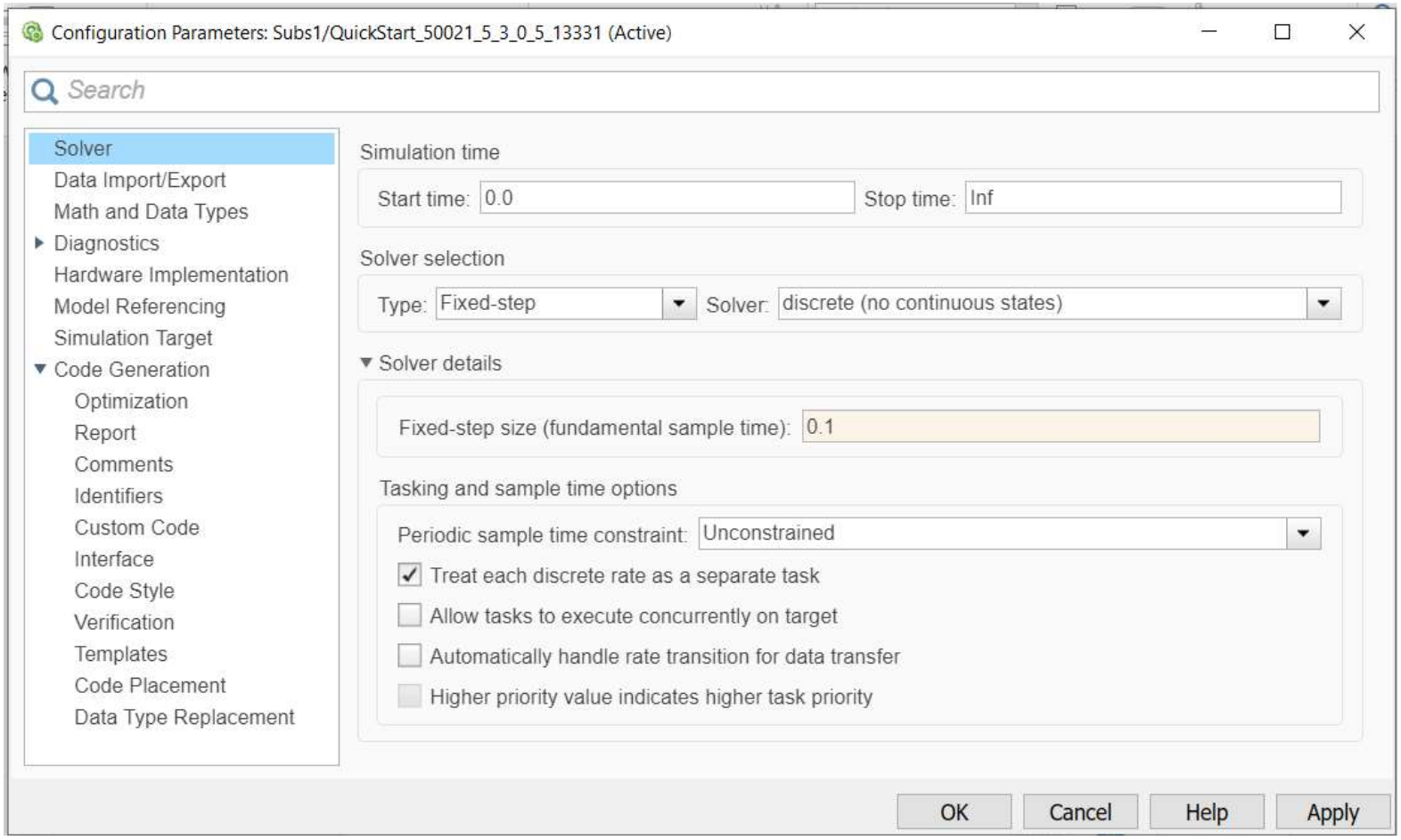
S-function builder, automatic SIL block…

5) Integrace

„Ruční“ integrace, testy integrace….

Generování aplikačního SW ze Simulinku

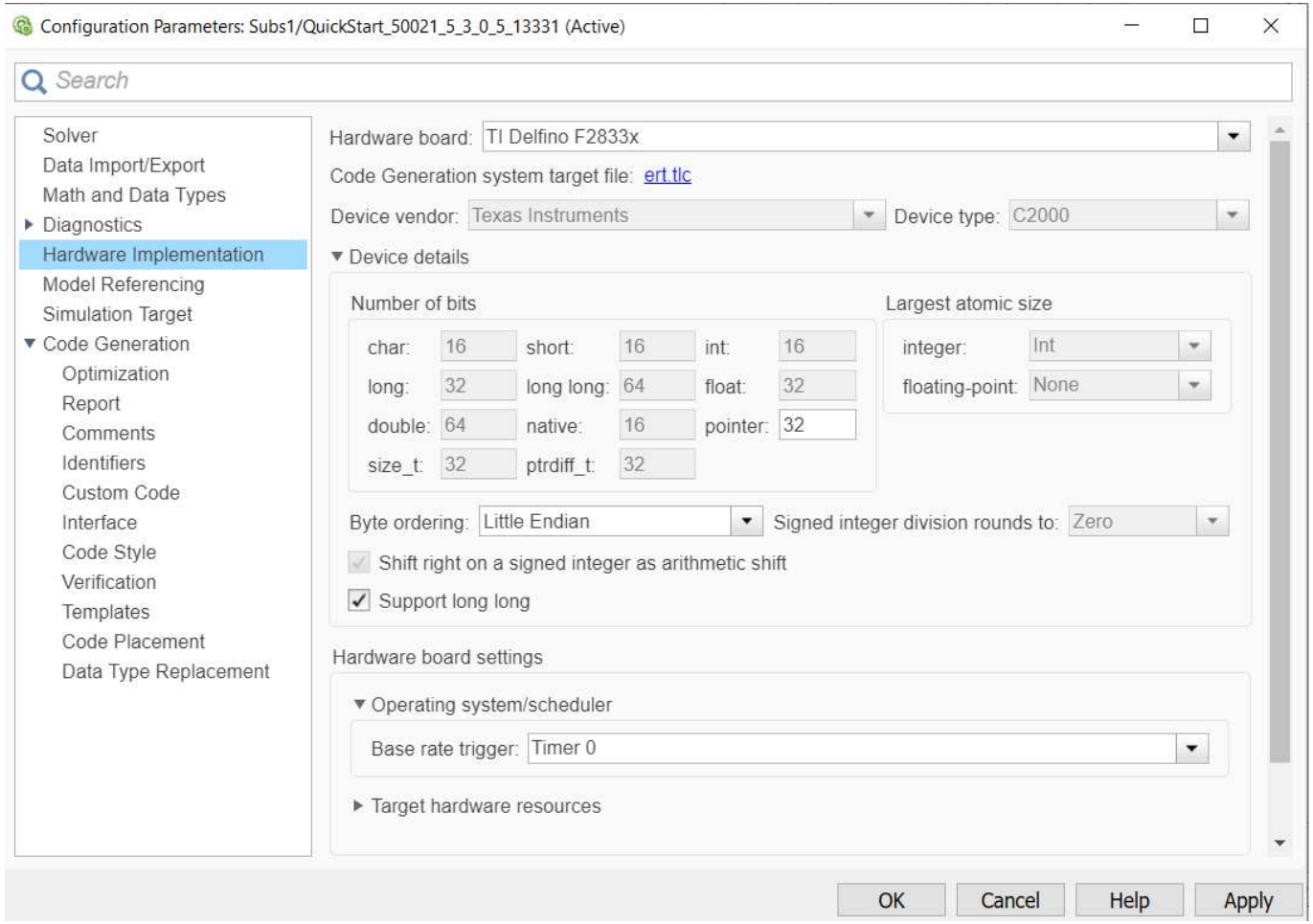
Konfigurace modelu pro generování kódu Solver



MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

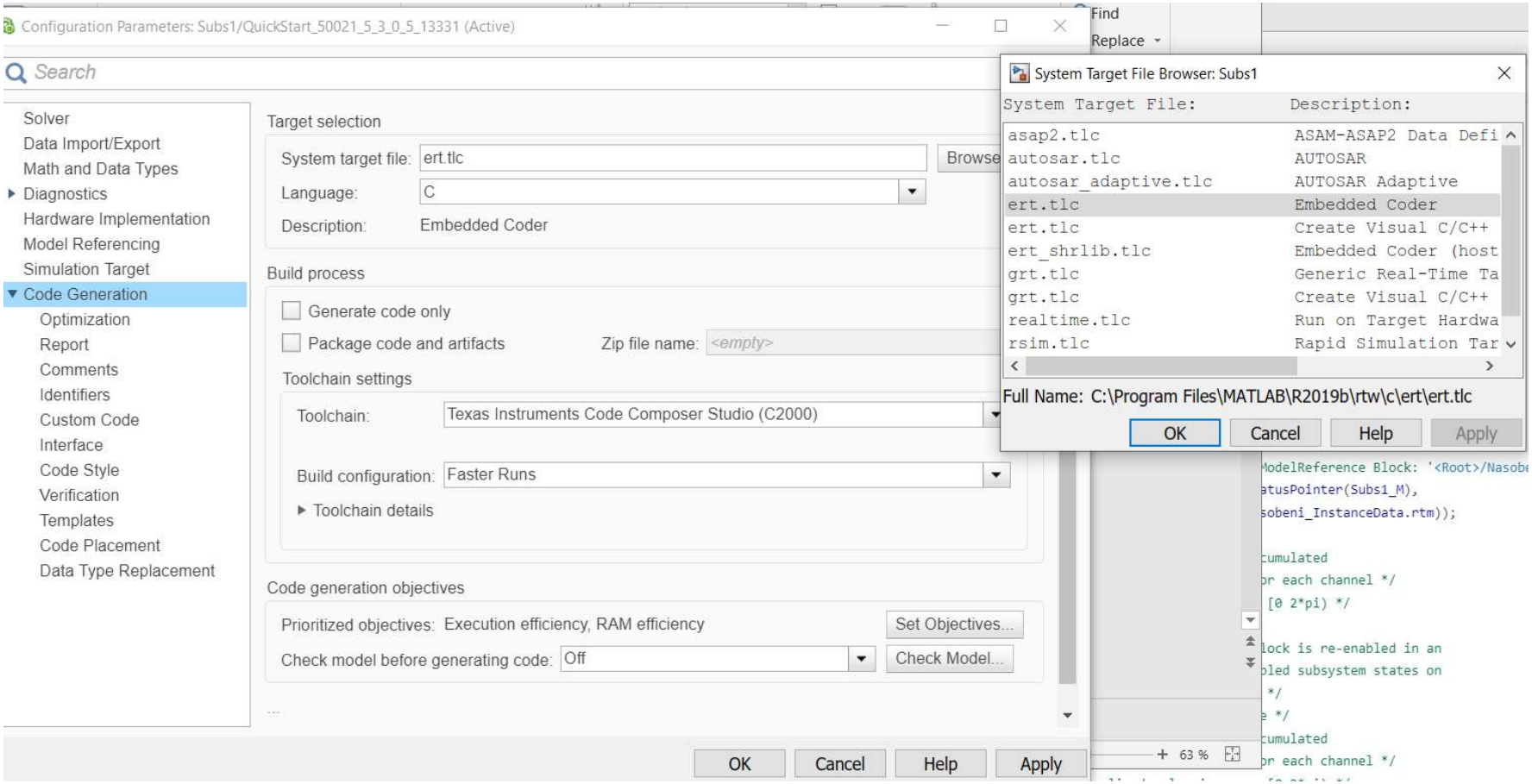
Generování aplikačního SW ze Simulinku

Konfigurace modelu pro generování kódu Hardware Implementation

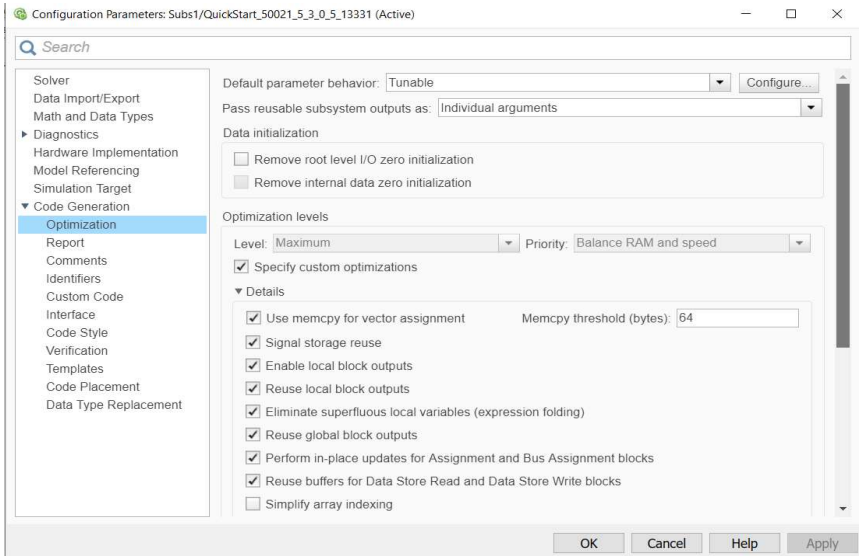


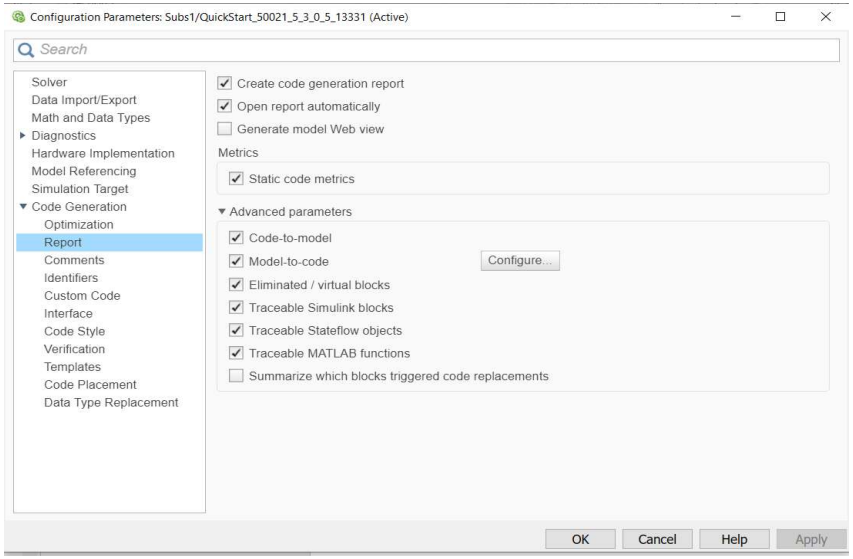
Generování aplikačního SW ze Simulinku

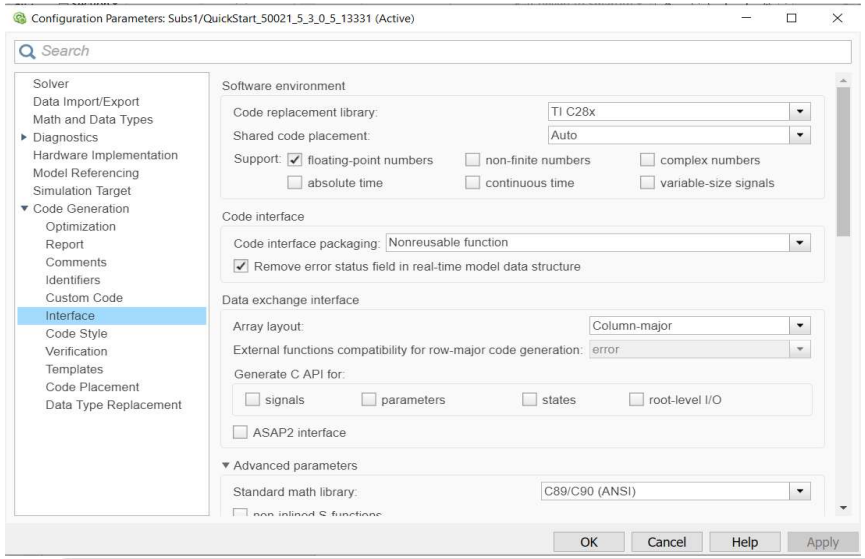
Konfigurace modelu pro generování kódu Code Generation (mnoho dalších nastavení)

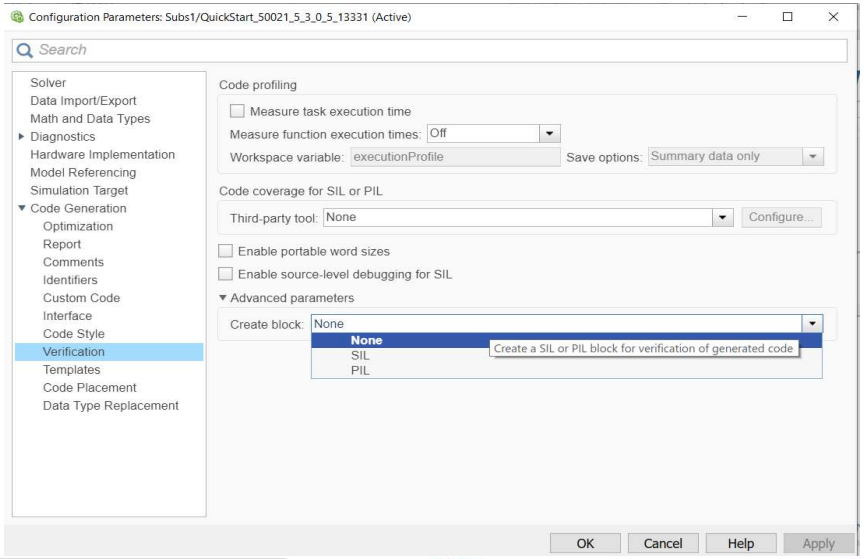


Generování aplikačního SW ze Simulinku



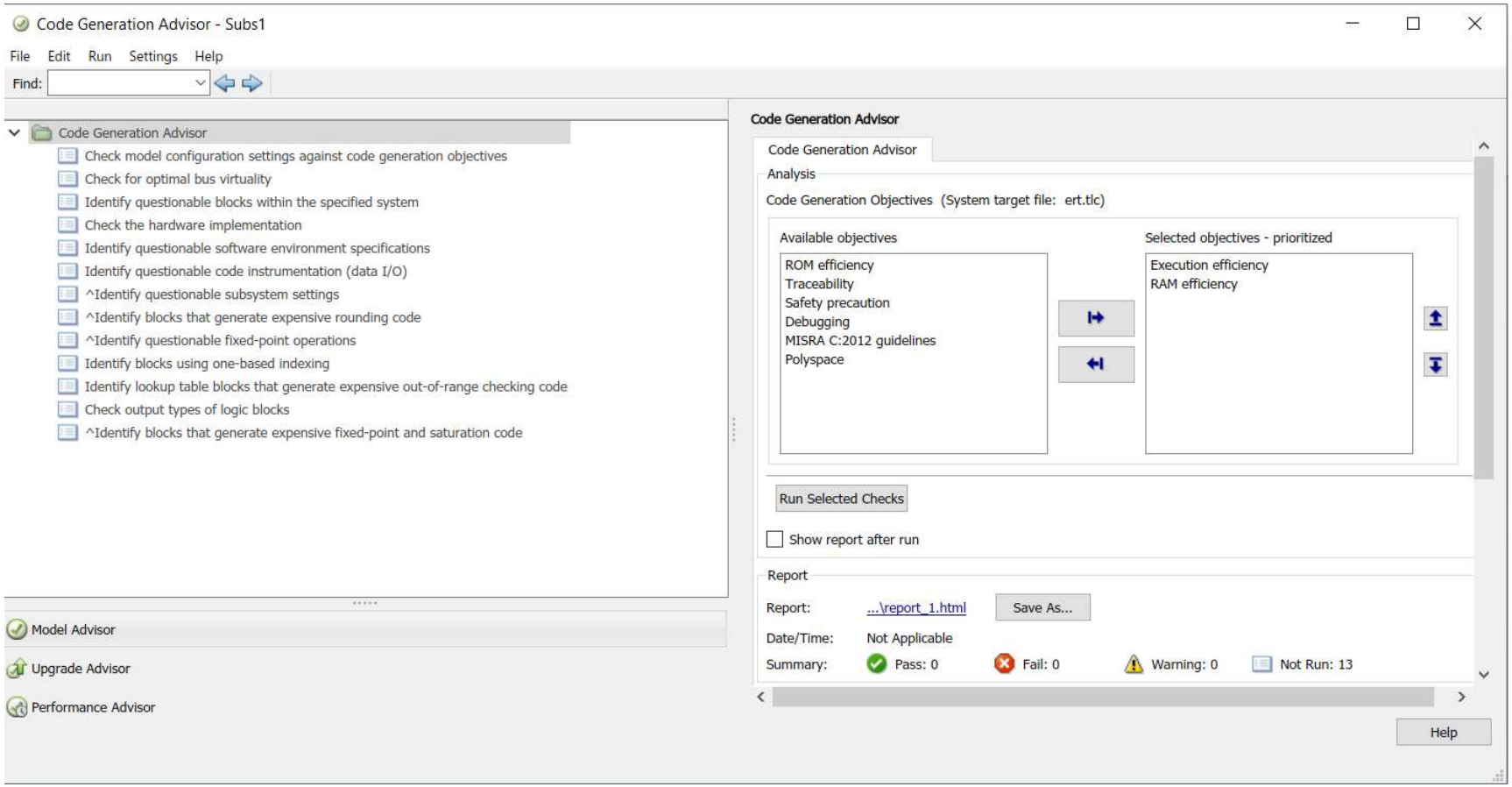






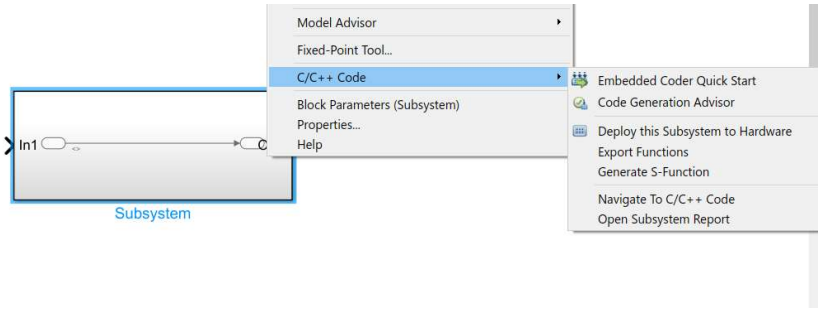
Generování aplikačního SW ze Simulinku

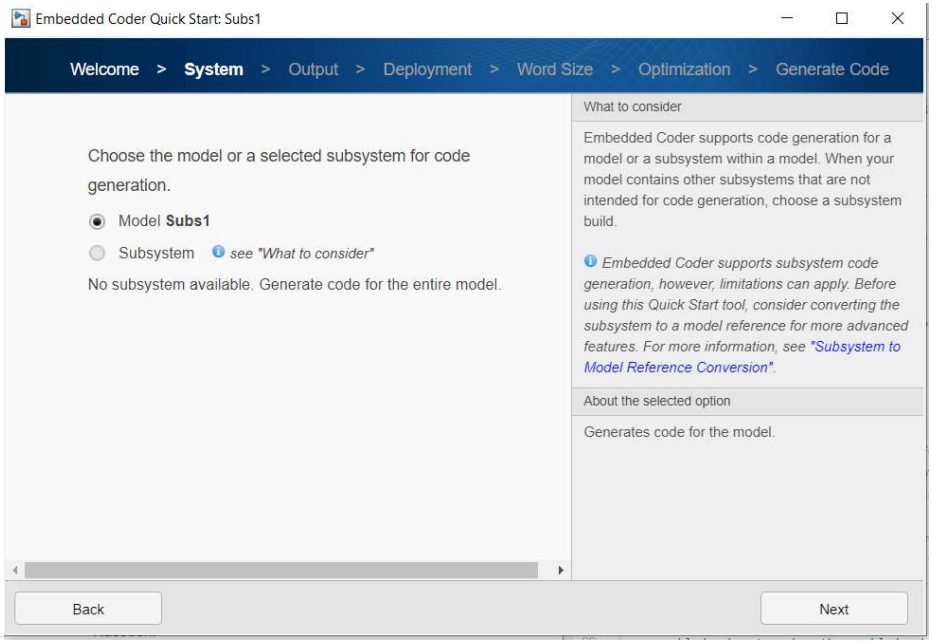
Konfigurace modelu pro generování kódu – Code generation advisor



Generování aplikačního SW ze Simulinku

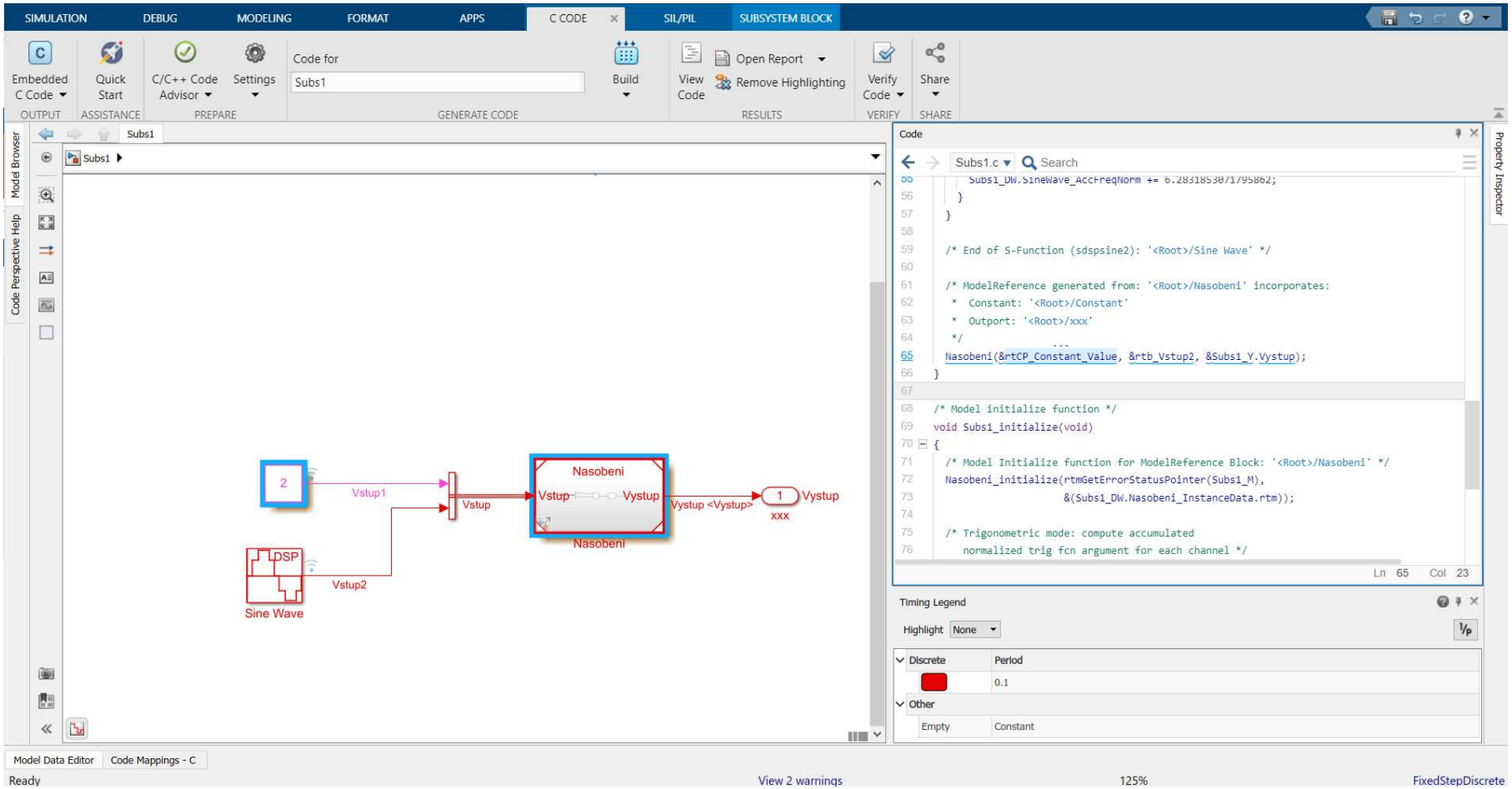
Generování kódu = pro jednoduché nastavení Embedded Coder Quick Start





Generování aplikačního SW ze Simulinku

Práce se Simulinkem a vygenerovaným kódem

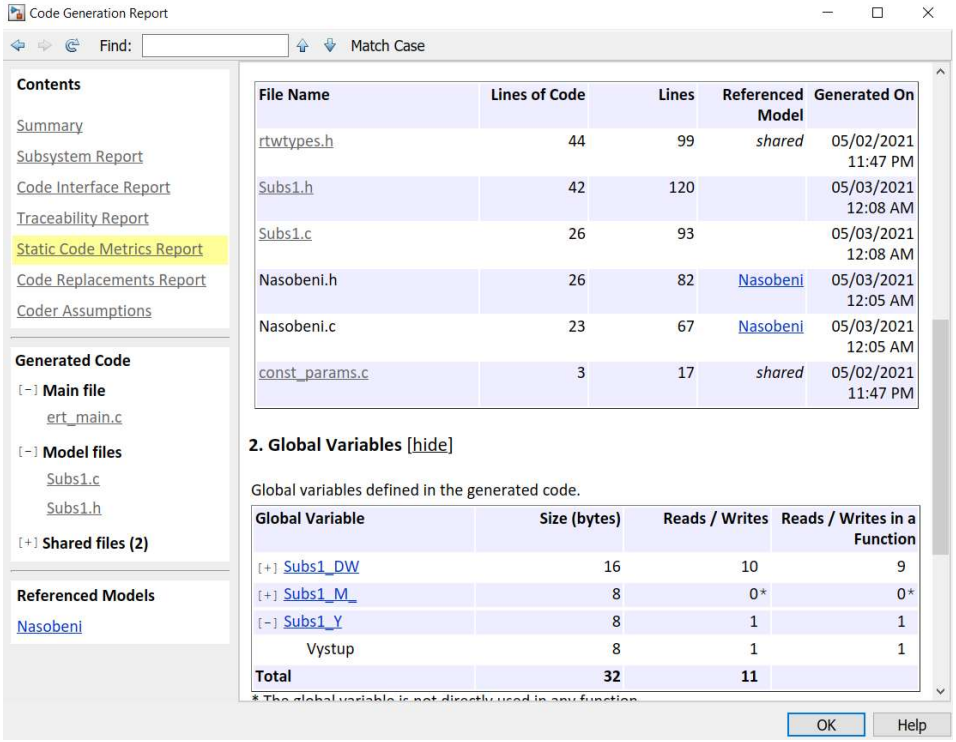


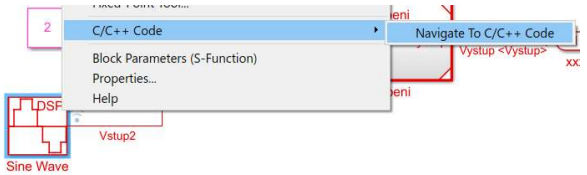
Report

Vygenerování HTML reportu

Report je ve formátu html provázaný s modelem pomocí odkazů :

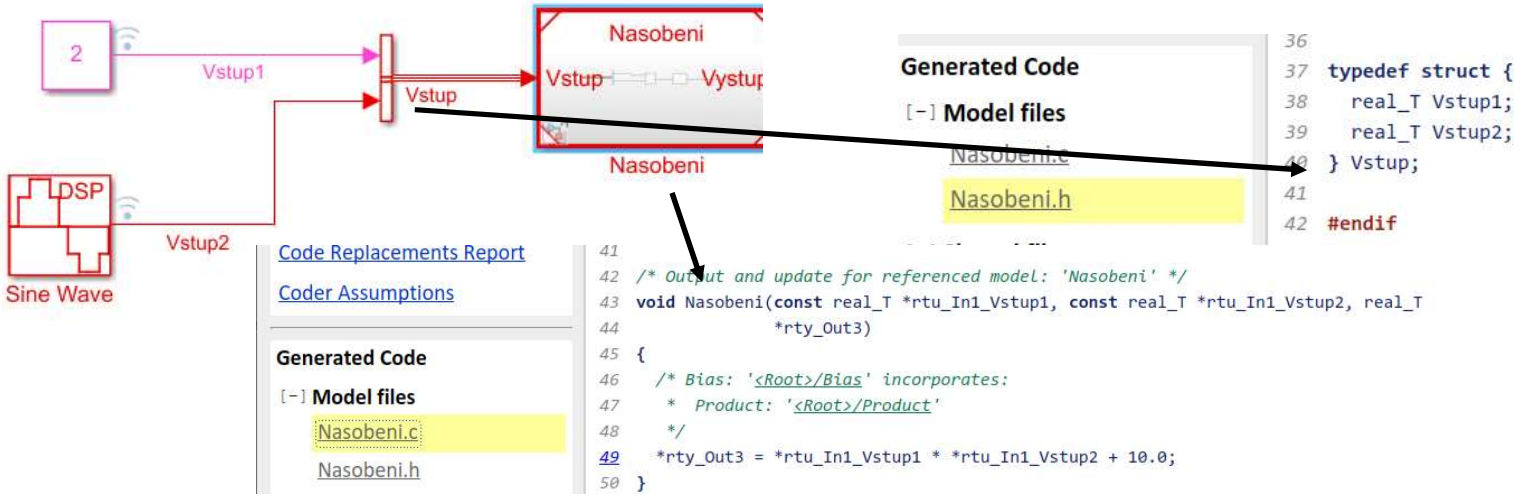
Model to Code a Code to Model



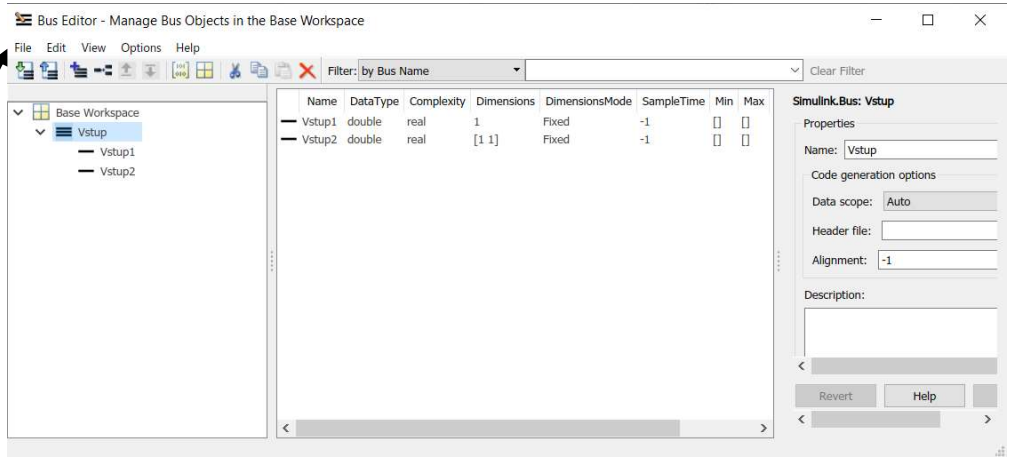


Názvy signálů a funkcí

Názvy signálů a funkcí – doporučuje se používat bus místo multiplexoru



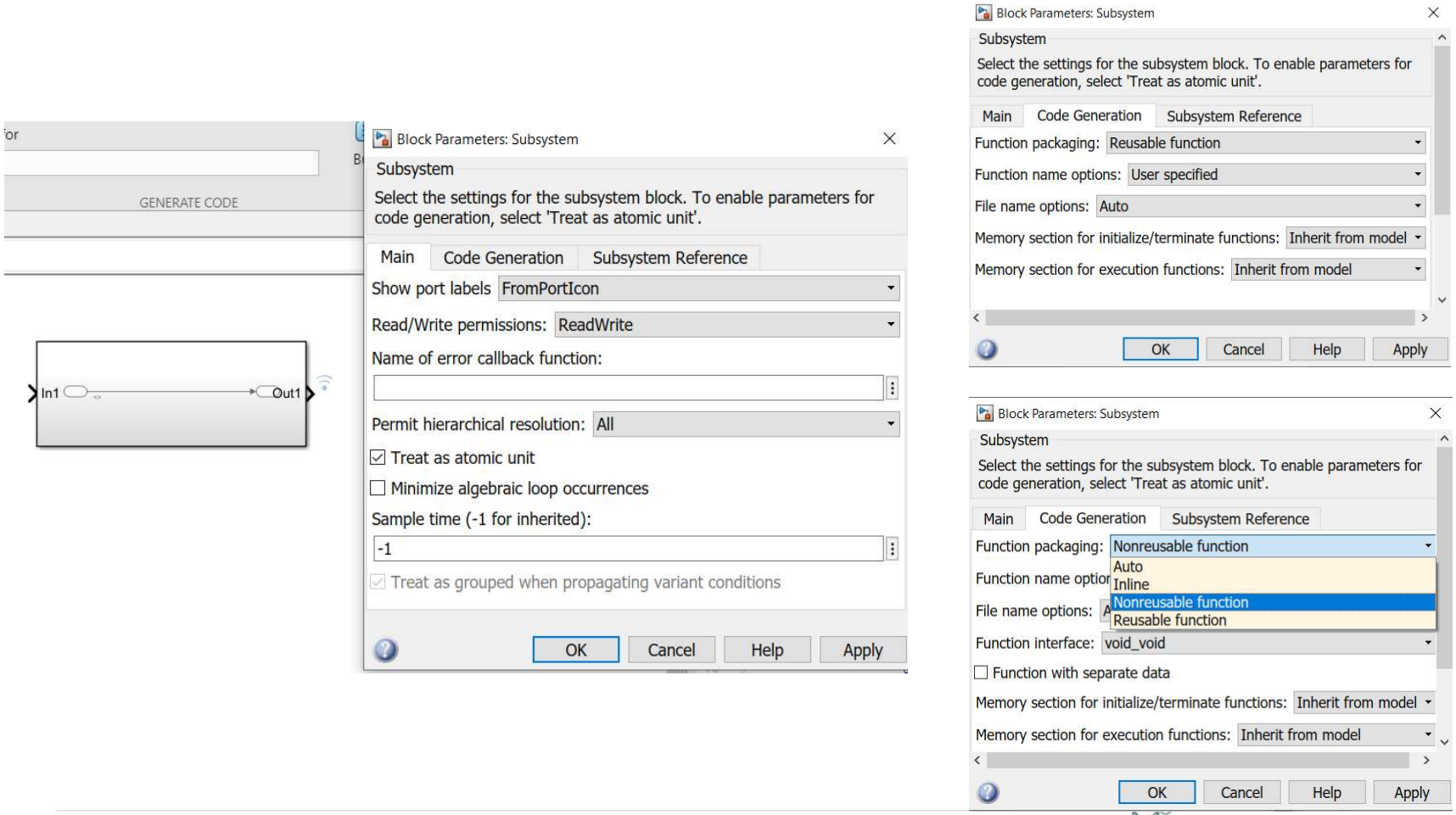
Můžeme používat Bus editor (spuštěn např. z Matlabu příkazem buseditor)



Generování subsystému

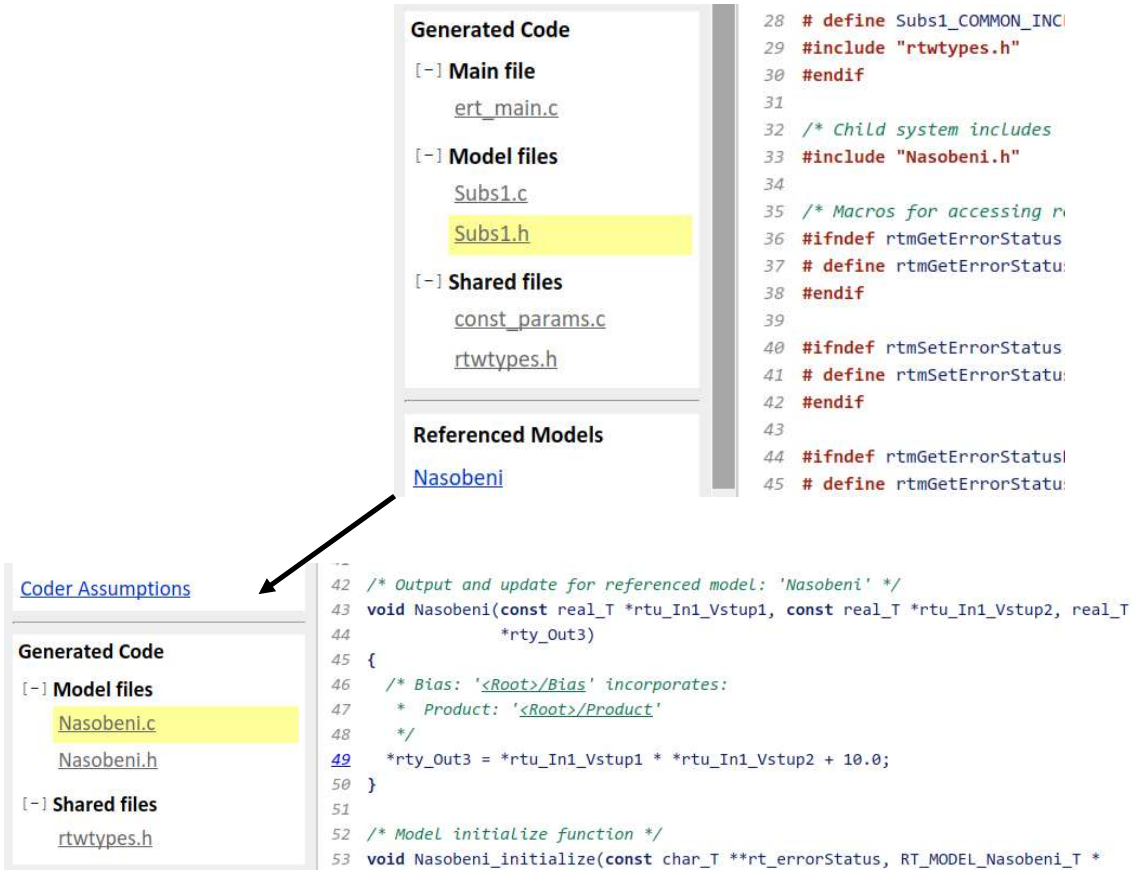
Parametry generování subsystému lze nastavit v Block parameters

Reusable function – pokud subsystém jako knihovna



Generování s referenčním modelem

Reference model = atomický uzavřený kód, který se snadno integruje, např. subsystém není implicitně atomický

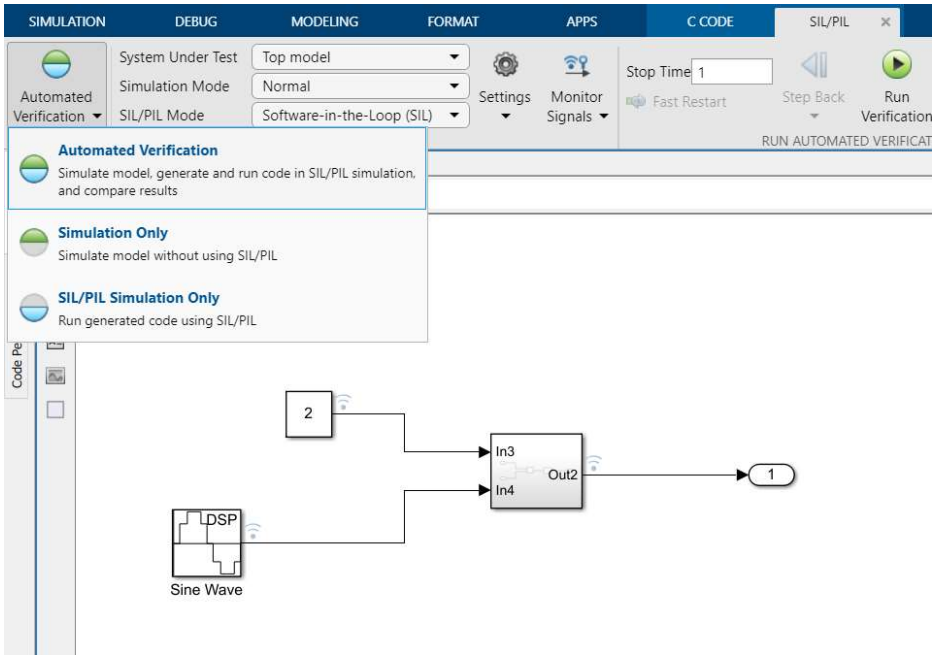


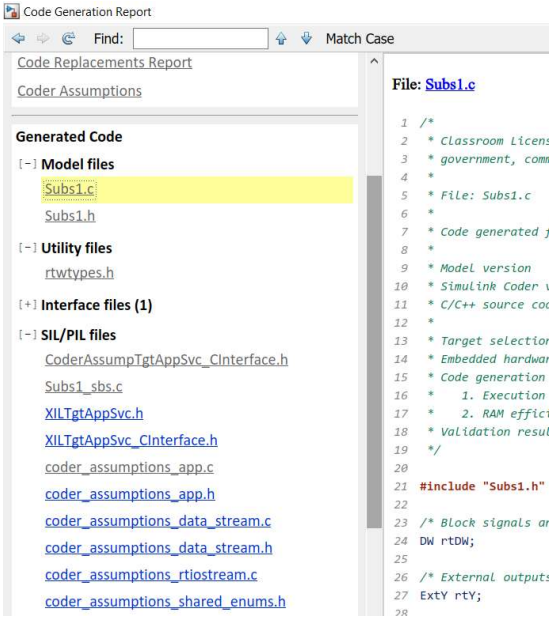
Testování SIL

Testování pomocí SIL

Otestování SW pomocí automatizované verifikace, SIL a Data Inspector

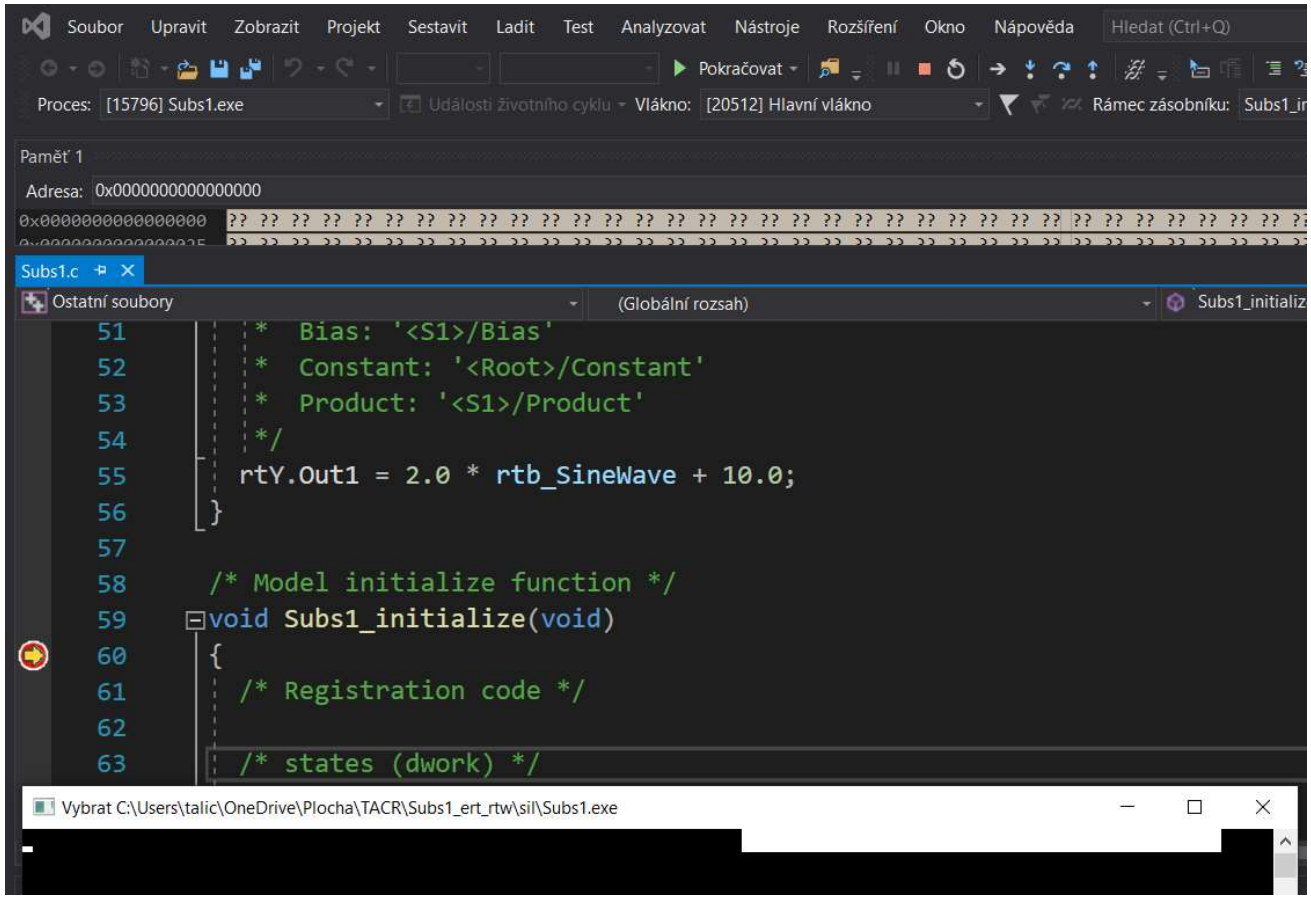
Simulink automaticky vygeneruje interface s vygenerovaným kódem subsystému





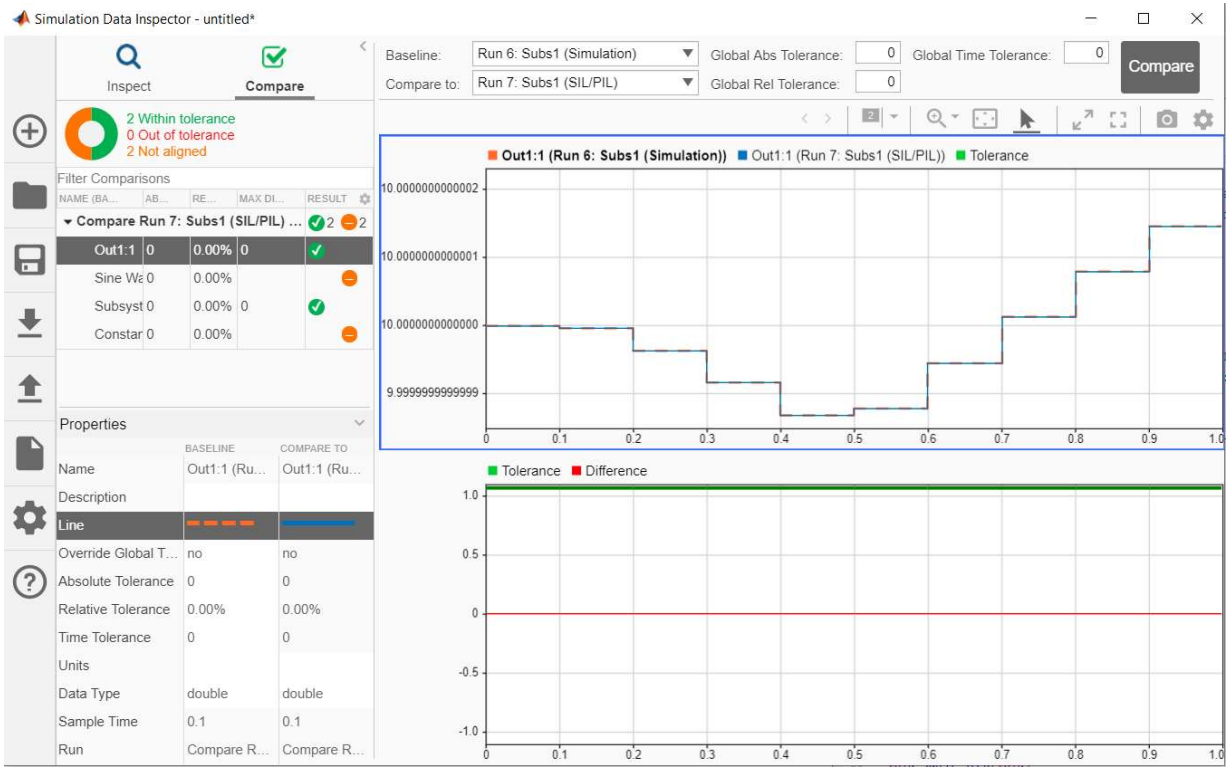
Testování pomocí SIL

Je vygenerován projekt, vytvořen spustitelný soubor a automaticky laděn ve Visual studiu, krok po kroku, výsledky posílány zpět do Simulinku do Data Inspect



Testování pomocí SIL

Otestování SW pomocí automatizované verifikace, SIL a Data Inspector



FAKULTA
ELEKTROTECHNICKÁ
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Příští přednáška

Cílové automatické generování kódu ze Simulinku

Jakub Talla

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

FAKULTA
ELEKTROTECHNICKÁ
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Děkuji za pozornost!

Adresa:
Univerzitní 26
306 14 Plzeň
Česká republika

Tel: +420 377 634 443
Fax: +420 377 634 402

Email: talic@kev.zcu.cz

www.rice.zcu.cz

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj pro inovace