FAKULTA
ELEKTROTECHNICKÁ
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Základy automatického generování kódu z Matlabu

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. 8

1) Úvod do generování kódu z Matlabu

- Základy generování kódu z Matlabu

2) Automatické generování kódu z Matlabu

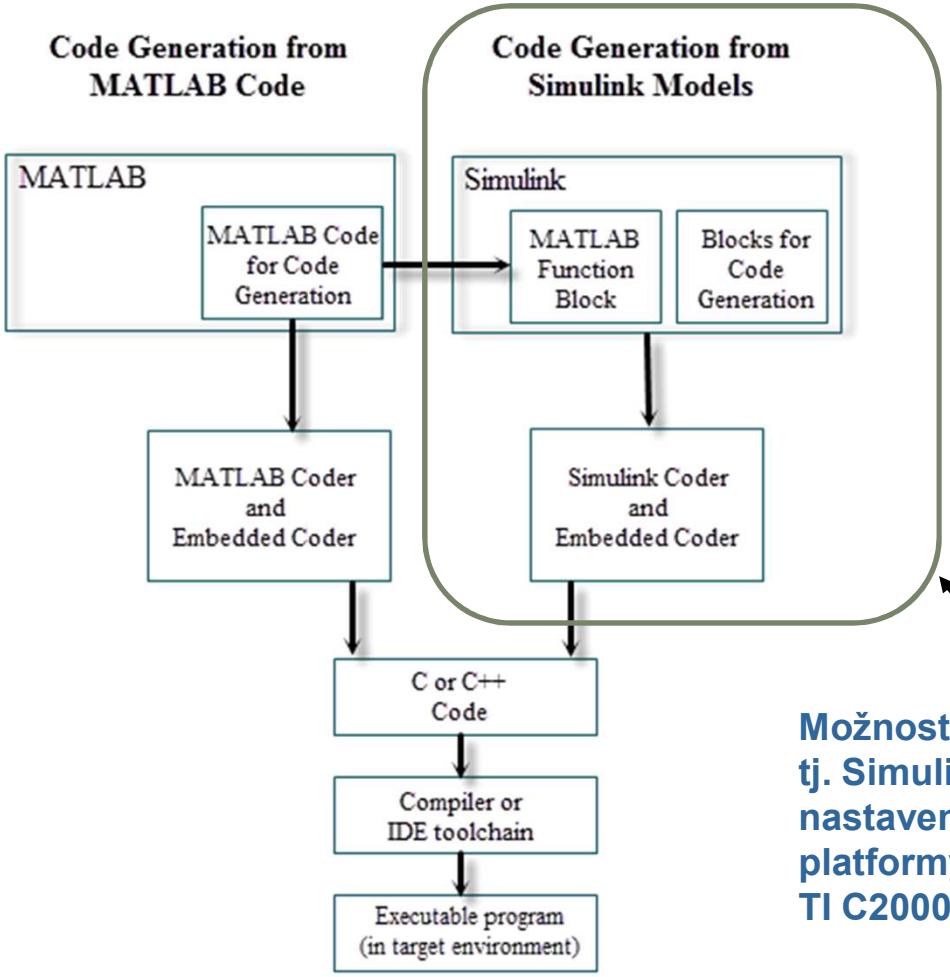
- Použitím příkazu codegen
- Použitím aplikace Matlab coder

3) Možnosti importu vlastního C kódu do Matlabu

- Použitím příkazu mex
- Použitím aplikace Matlab coder

Základní generování C/C++ kódu z Matlab/Simulink

Matlab + Simulink



Možnost použít „targety“ tj. Simulink bloky a nastavení HW cílové platformy (např. procesory TI C2000 série)

Generování C kódu z Matlabu

Generování C kódu z Matlabu

Automatické generování C kódu z Matlabu

Využívá Matlab coder toolbox

Slouží převážně k:

- Akceleraci výpočtů na PC (vytvoření MEX souboru)
- Generování částí aplikačního SW pro PC (vytvoření .C nebo .lib)
- Generování zdrojového kódu aplikačního SW pro embedded zařízení (mikrokontroléry….)

Nutné části:

- Matlab
- Matlab coder
- Podporovaný C/C++ compiler (Visual studio, MinGW,….)

Generování C kódu z Matlabu

Možnosti testování a generování C kódu v Matlabu

codegen – slouží ke generování kódu, umí vygenerovat zdrojové kódy, knihovny, ale i spustitelné mex (z Matlab) a .exe

Možnost kompilace:

- 1) Z příkazové řádky: *codegen FceGen2 -args {x1in, x2in} -nargout 2*
- 2) Pomocí Matlab coder aplikace

Mex – slouží k vytvoření a zkompilování do mex souboru vlastní projekt ze zdrojových souborů C/C++

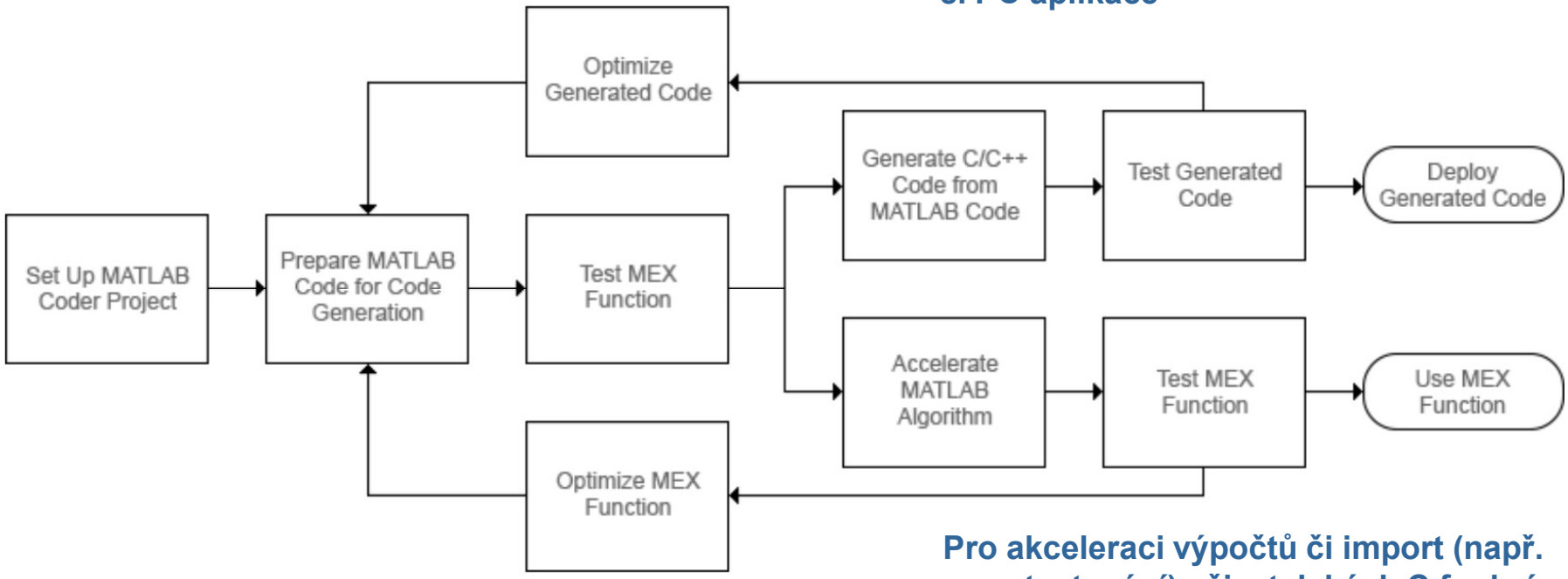
Možnost kompilace:

- 1) Z příkazové řádky: *mex parametry* (nutné nadefinovat datové rozhraní s Matlabem ve zdrojovém kódu)
- 2) Pomocí Matlab coder aplikace

Generování C kódu z Matlabu

Automatické generování C kódu z Matlabu

Pro generování kódu pro mikroprocesory či PC aplikace

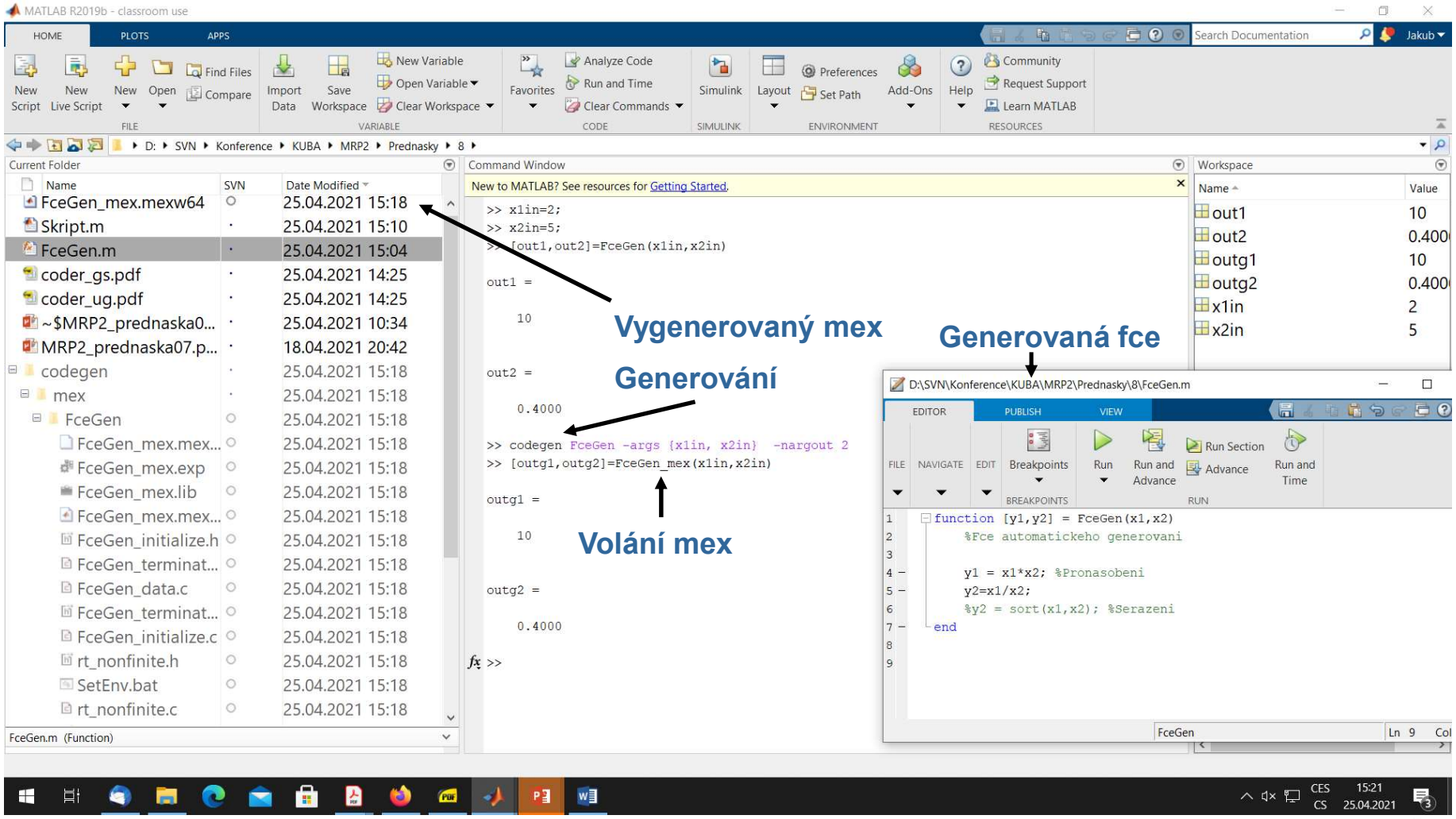


Pro akceleraci výpočtů či import (např. pro otestování) uživatelských C funkcí

Codegen - generování C kódu z Matlab scriptu/funkce

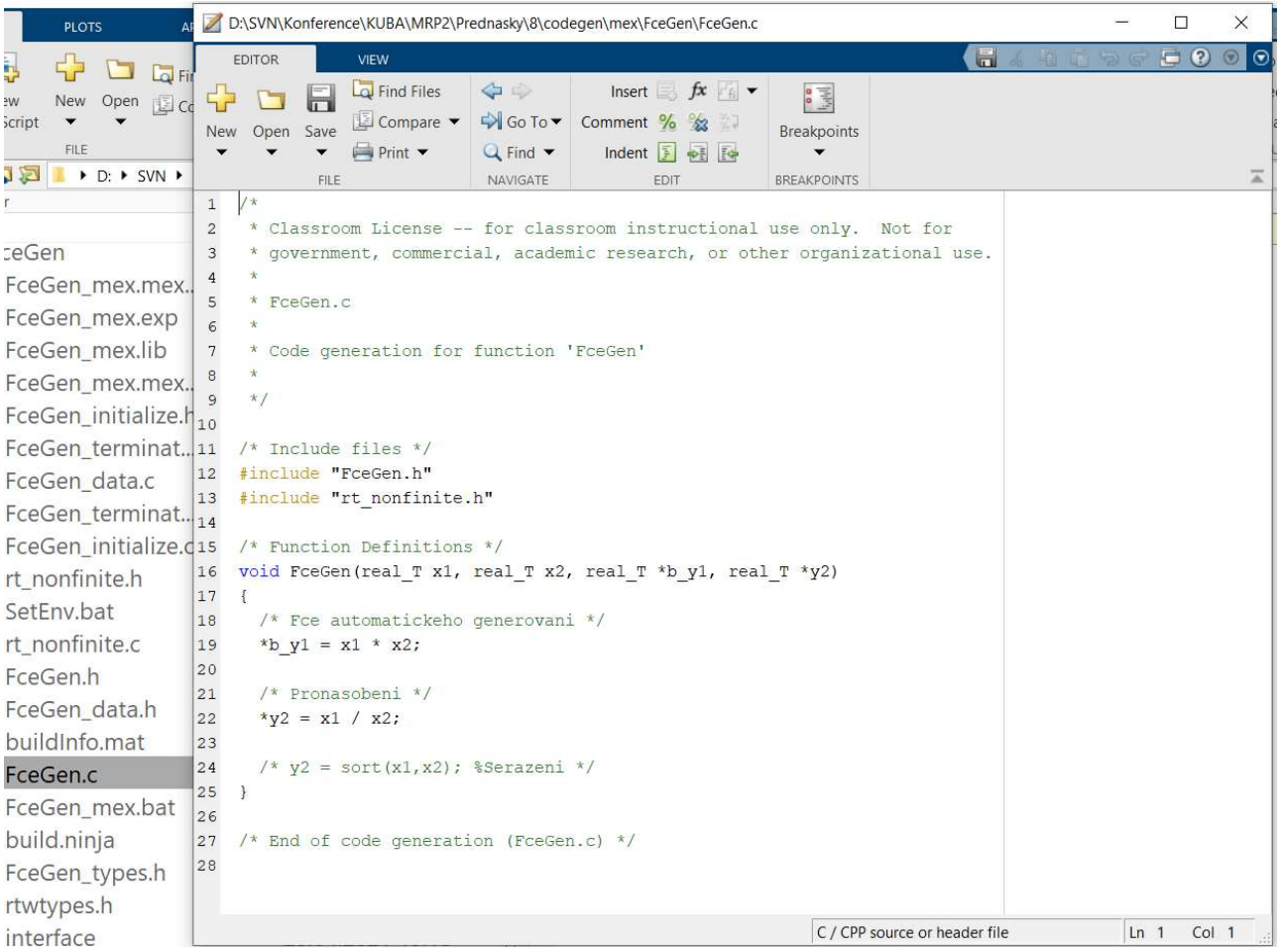
Generování C kódu z příkazové řádky

Codegen příklad: codegen FceGen2 -args {x1in, x2in} -nargout 2



Generování C kódu z příkazové řádky

Příklad vygenerovaného zdrojového C kódu funkce FceGen



Generování C kódu z příkazové řádky

Pro vygenerování kódu je často nutná řada nastavení (rozměry a datové typy vstupů, mex/exe/lib apod.), které se zadávají jako parametry funkce codegen

```
COMMON OPTIONS:

-args ARGS  Specify the types that the generated code should
            accept.  ARGS is a cell array specifying the type of
            each function argument.  (Elements are converted to
            types using coder.typeof.)

-nargout NARGOUT Specify the number of output arguments for the given
            function. If using VARARGOUT, this is required.

-config:mex  Generate a MEX function (default)
-config:lib  Generate a C/C++ static library
-config:dll  Generate a C/C++ dynamic library
-config:exe  Generate a C/C++ executable
-config:hdl  Generate VHDL/Verilog code

-config CONFIG  Specify a custom configuration object (use coder.config
            to create one).

-double2single CONFIG Specify a doubles to singles configuration object
            to convert double-precision MATLAB code to single-precision
            MATLAB code (use coder.config to create one).

-float2fixed CONFIG Specify a fixpt configuration object(use
            coder.config to create one).

-o OUTPUT   Specify the name of the output.  codegen adds a
            platform-specific extension to this name depending
            on the output type.
```

Generování C kódu z příkazové řádky

Další nastavení se provádí příkazem *coder*. Většinu nastavení lze „naklikat“ v aplikaci Matlab coder.

```
cfg = coder.config('lib');
```

Příkaz *coder* najde uplatnění i v konfiguraci Matlab scriptu/funkci

Například definování proměnné h s dynamickou velikostí jednorozměrného pole o až 10 prvcích apod. (v Matlabu není nutné definovat velikost proměnných)

```
cfg = coder.config('lib');
coder.varsize('h', [1 10]);
h=…
```

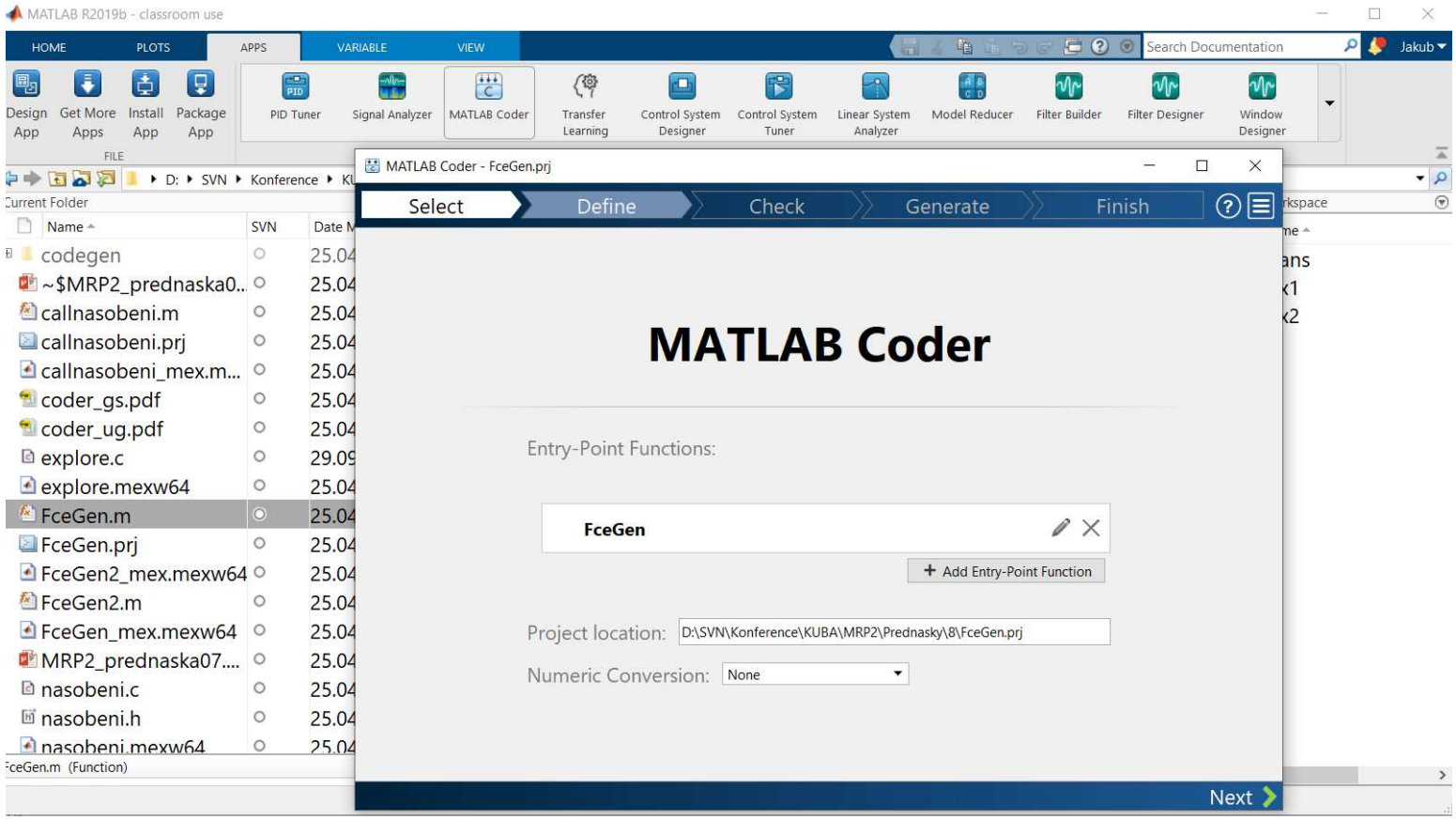
Příkazem *coder* je také možné volat např. uživatelské c funkce v rámci vygenerovaného kódu

```
coder.updateBuildInfo('addSourceFiles','nasobeni.c');
coder.cinclude('nasobeni.h‘);

y = coder.ceval('nasobeni', x1, x2);
```

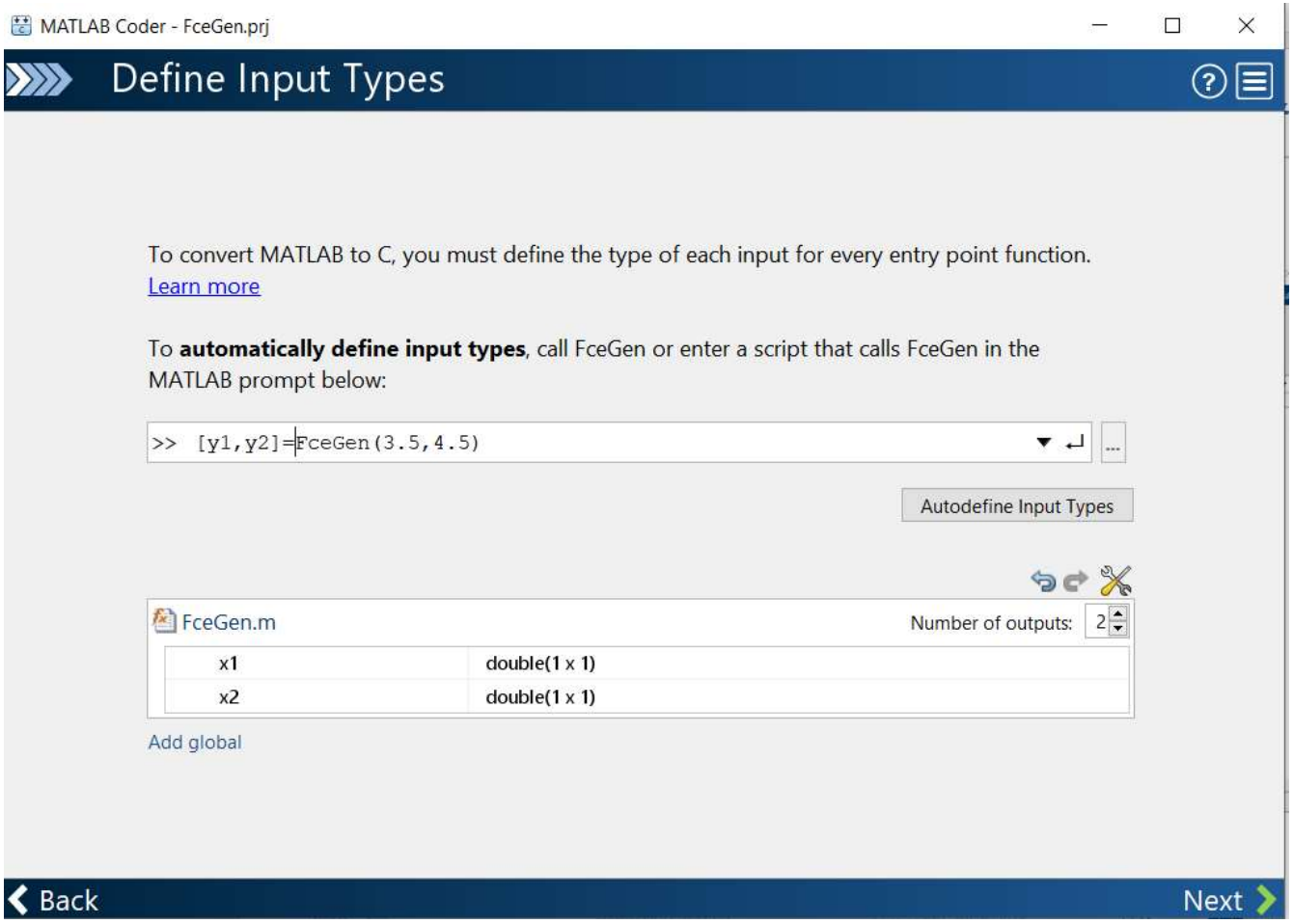
Generování C kódu Matlab Coder

Generování kódu pomocí Matlab Coder prostředí



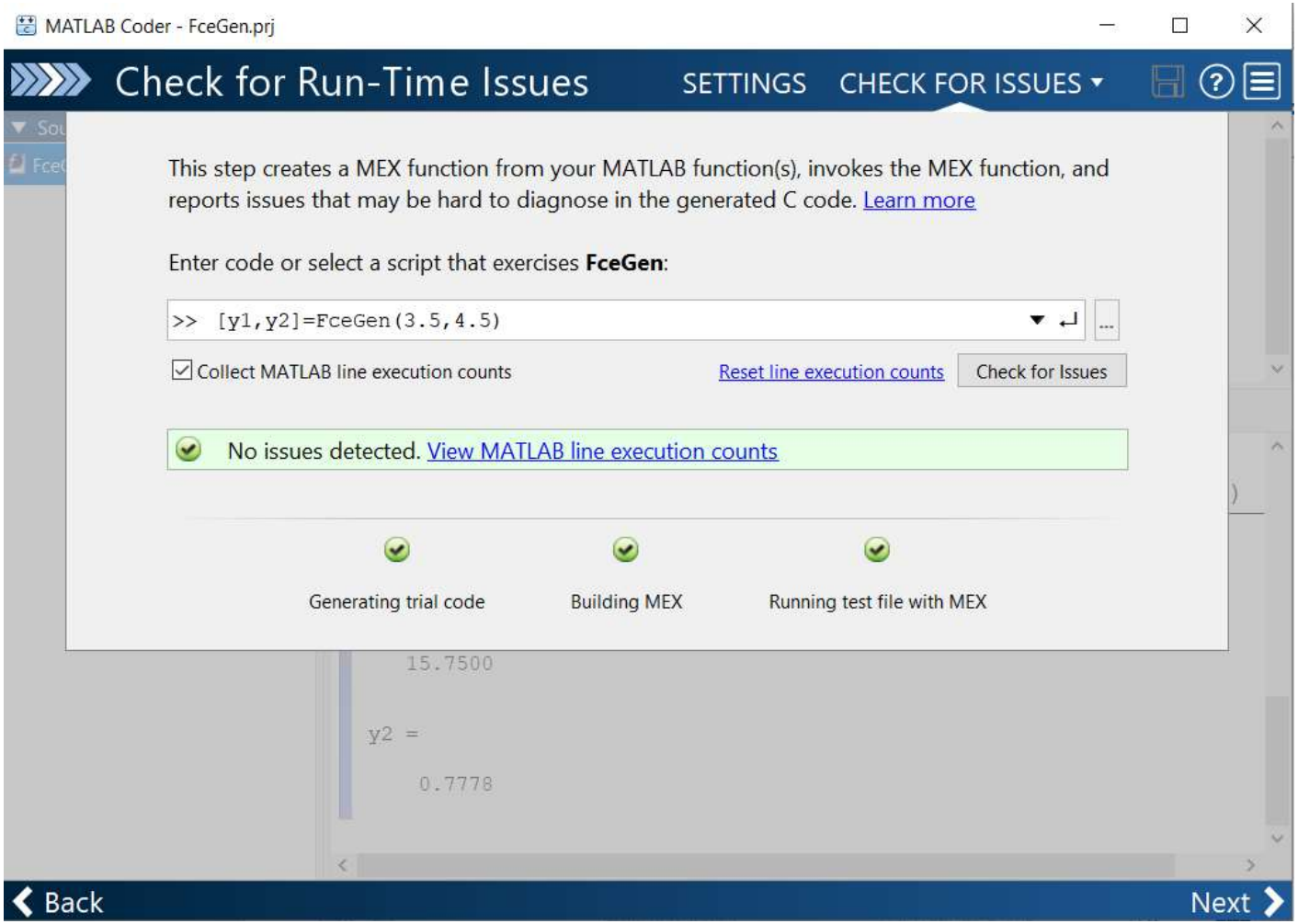
Generování C kódu Matlab Coder

Generování kódu pomocí Matlab Coder prostředí - definování vstupů



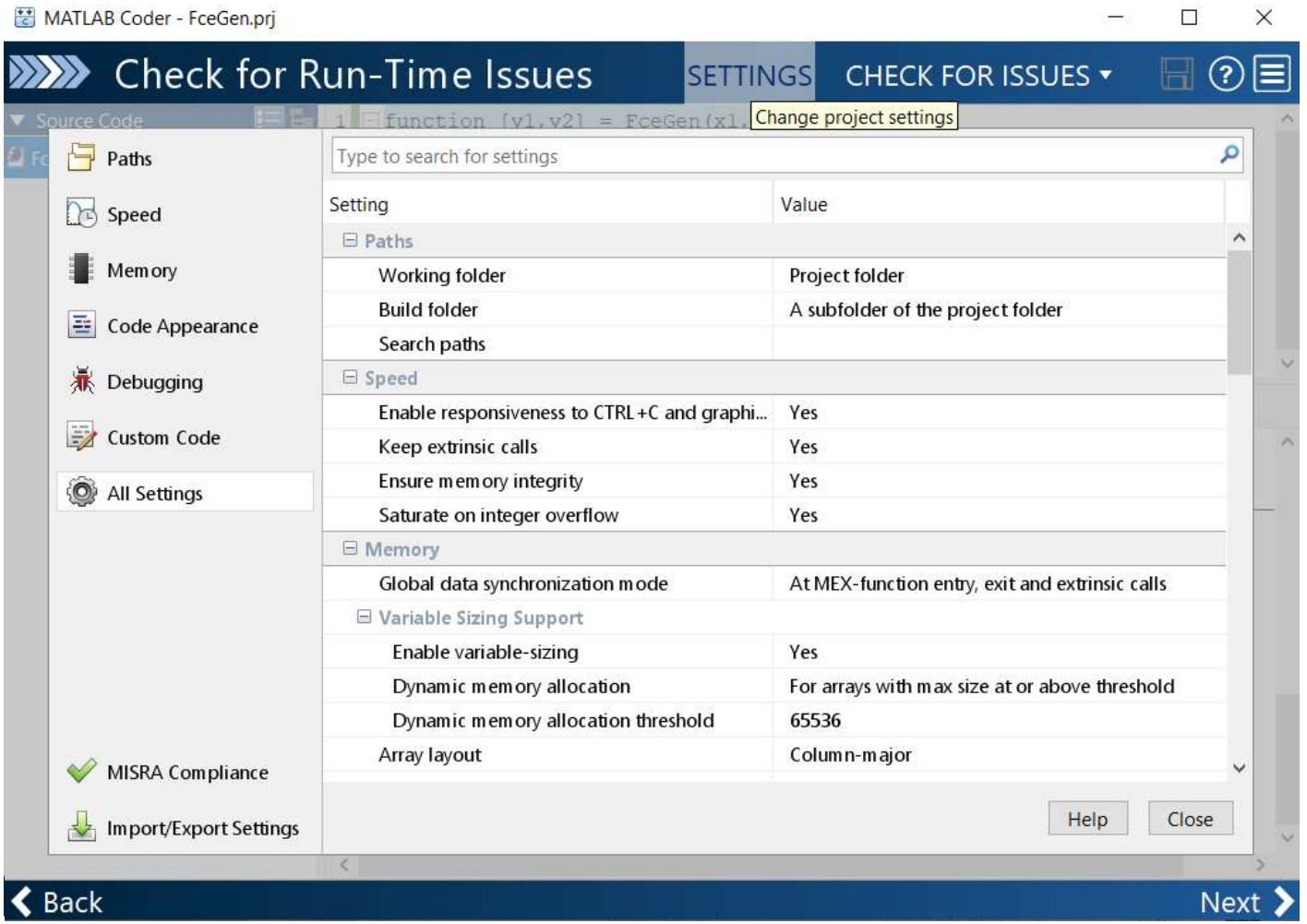
Generování C kódu Matlab Coder

Vytvoření mex a otestování funkce (rozdílu mezi mex a Matlab kódem)



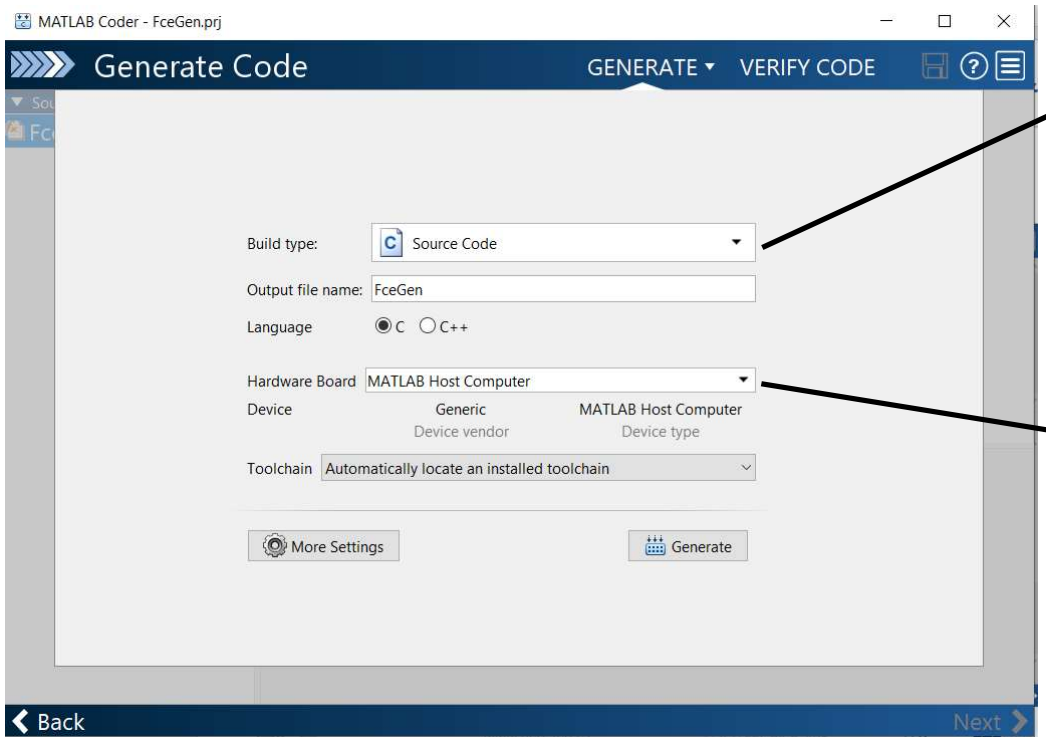
Generování C kódu Matlab Coder

Další konfigurace projektu

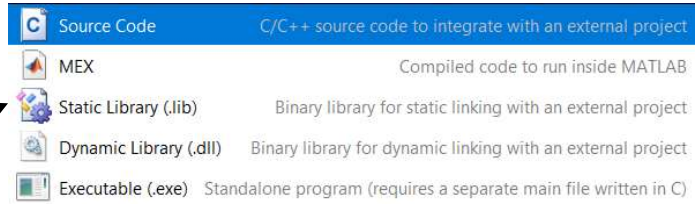


Generování C kódu Matlab Coder

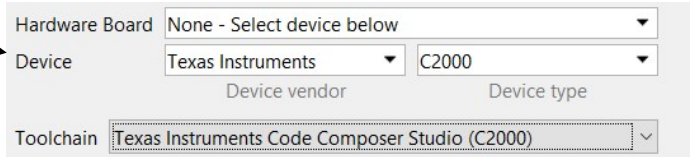
Generování projektu



Varianty implementace

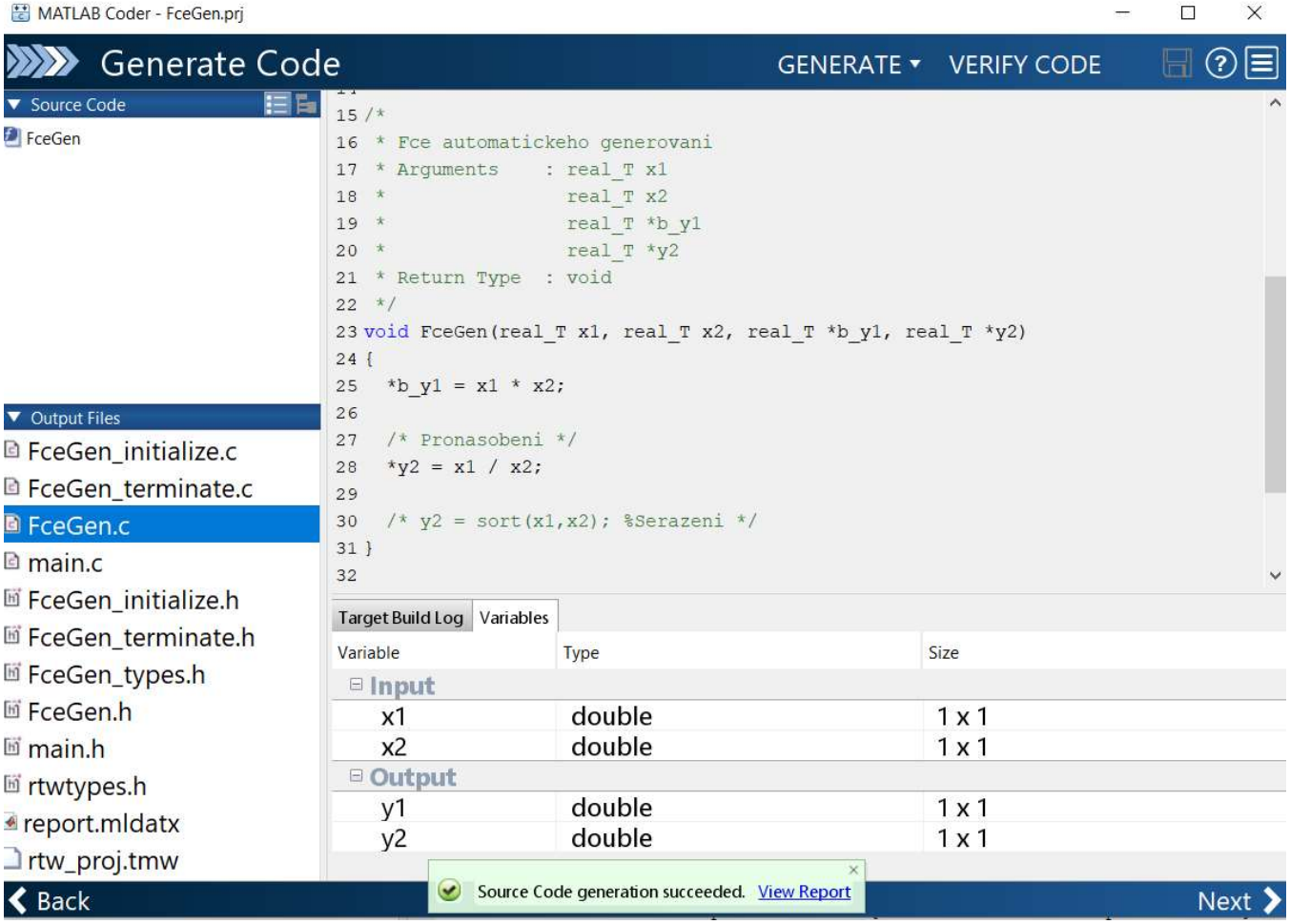


Varianty cílové platformy (ovlivňují hlavně velikost int apod.)



Generování C kódu Matlab Coder

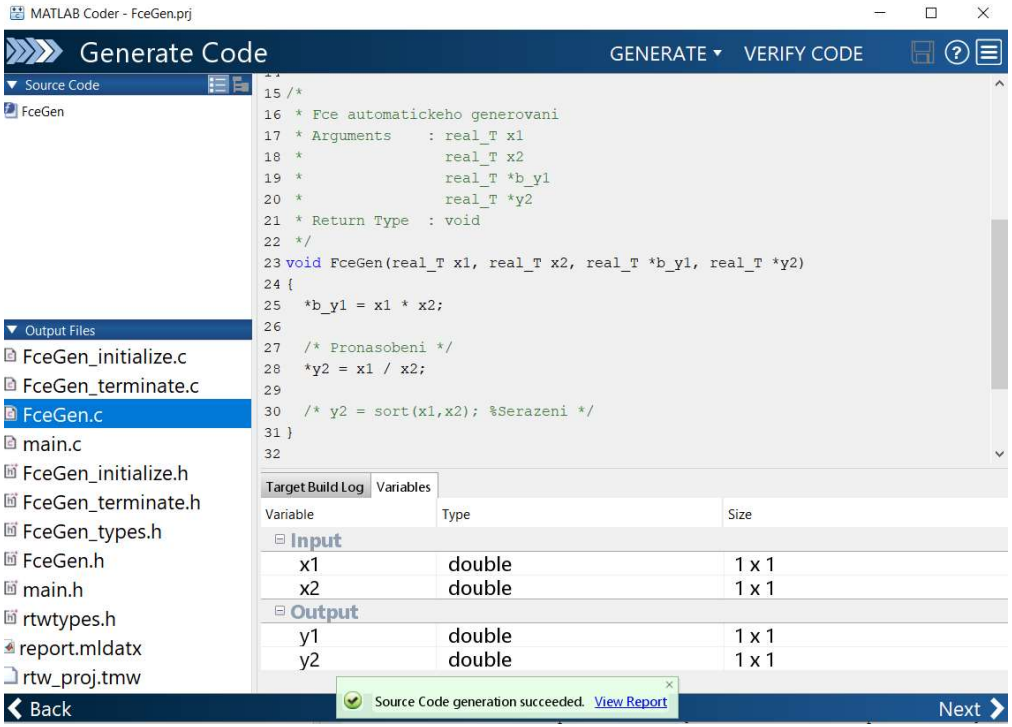
Vygenerovaný projekt



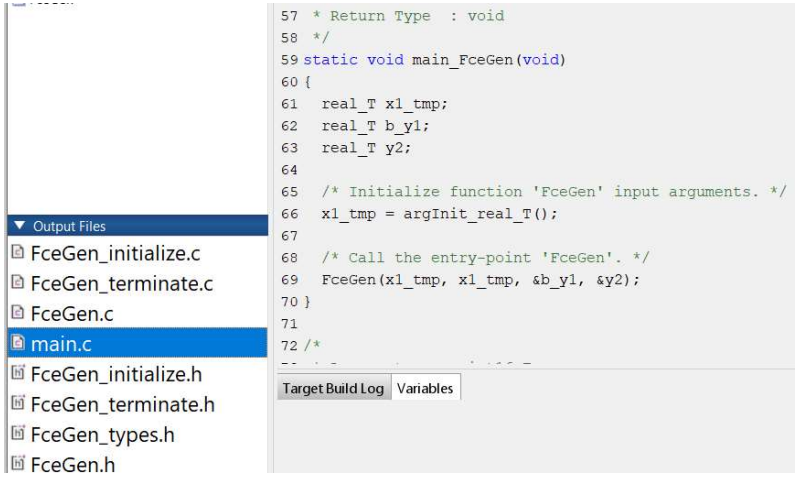
Generování C kódu Matlab Coder

Vygenerovaný projekt

FceGen.c vlastní tělo funkce



main.c volání funkce



Import a testování vlastní C funkce v Matlabu

Import C funkce do Matlabu

Tvorba MEX (Matlab EXecutable) z vlastního C kódu

mex příkaz umožňuje vytvořit z prostředí Matlab spustitelný mex/mexw64 soubor

Musí mít nadefinované rozhraní C a Matlab tj. datové typy, jejich velikosti apod. (Podobně jako v Simulink S-funkce)

Gateway funkce s definicí vstupů

a voláním uživatelské funkce

Uživatelská funkce

```c
#include "mex.h"

/* The computational routine */
void arrayProduct(double x, double *y, double *z, mwSize n)
{
    mwSize i;
    /* multiply each element y by x */
    for (i=0; i<n; i++) {
        z[i] = x * y[i];
    }
}
```

```c
/* The gateway function */
void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[])
{
    double multiplier;              /* input scalar */
    double *inMatrix;               /* 1xN input matrix */
    size_t ncols;                   /* size of matrix */
    double *outMatrix;              /* output matrix */

    /* check for proper number of arguments */
    if(nrhs!=2) {
        mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nrhs","Two inputs required.");
    }
    if(nlhs!=1) {
        mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs","One output required.");
    }
    /* make sure the first input argument is scalar */
    if( !mxIsDouble(prhs[0]) ||
         mxIsComplex(prhs[0]) ||
         mxGetNumberOfElements(prhs[0])!=1 ) {
        mexErrMsgIdAndTxt("MyToolbox:arrayProduct:notScalar","Input multiplier must be a scalar.");
    }

    /* make sure the second input argument is type double */
    if( !mxIsDouble(prhs[1]) ||

    /* call the computational routine */
    arrayProduct(multiplier,inMatrix,outMatrix,(mwSize)ncols);
}
```

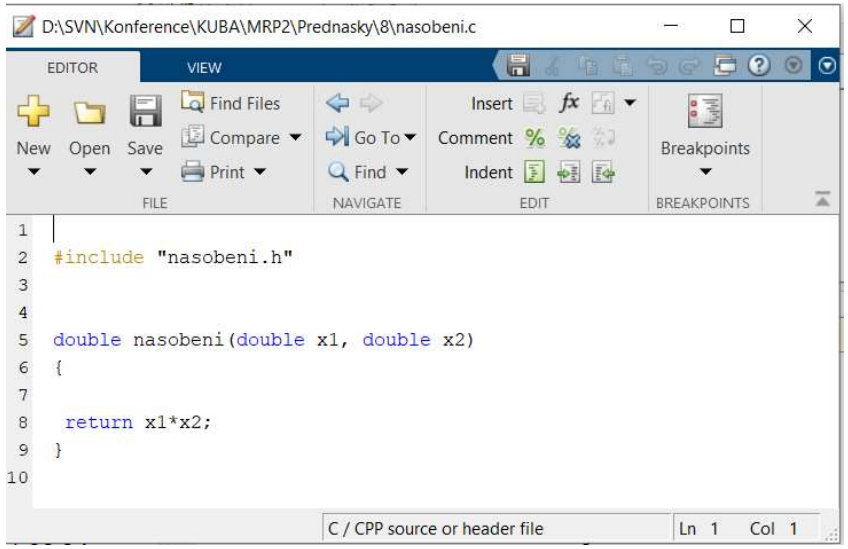
Import C funkce do Matlabu

Tvorba MEX z vlastního C pomocí Matlab coder

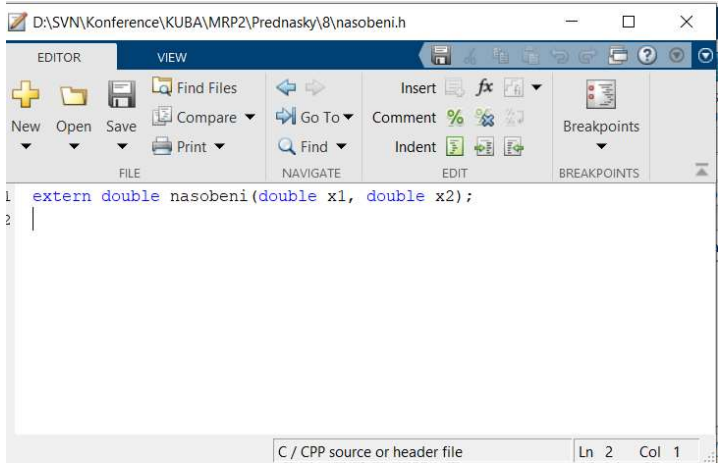
Tvorbu mex z vlastního C kódu lze velmi usnadnit za pomocí Matlab coder. Často navíc chceme porovnat funkci v C s funkcí v Matlabu (cosimulací či Software In the Loop).

V podstatě vytvoříme v Matlabu funkci, která bude volat naší C funkci a pro tu nakonfigurujeme interface s Matlabem a vygenerujeme kód

Uživatelská funkce nasobeni v modulu nasobeni.c



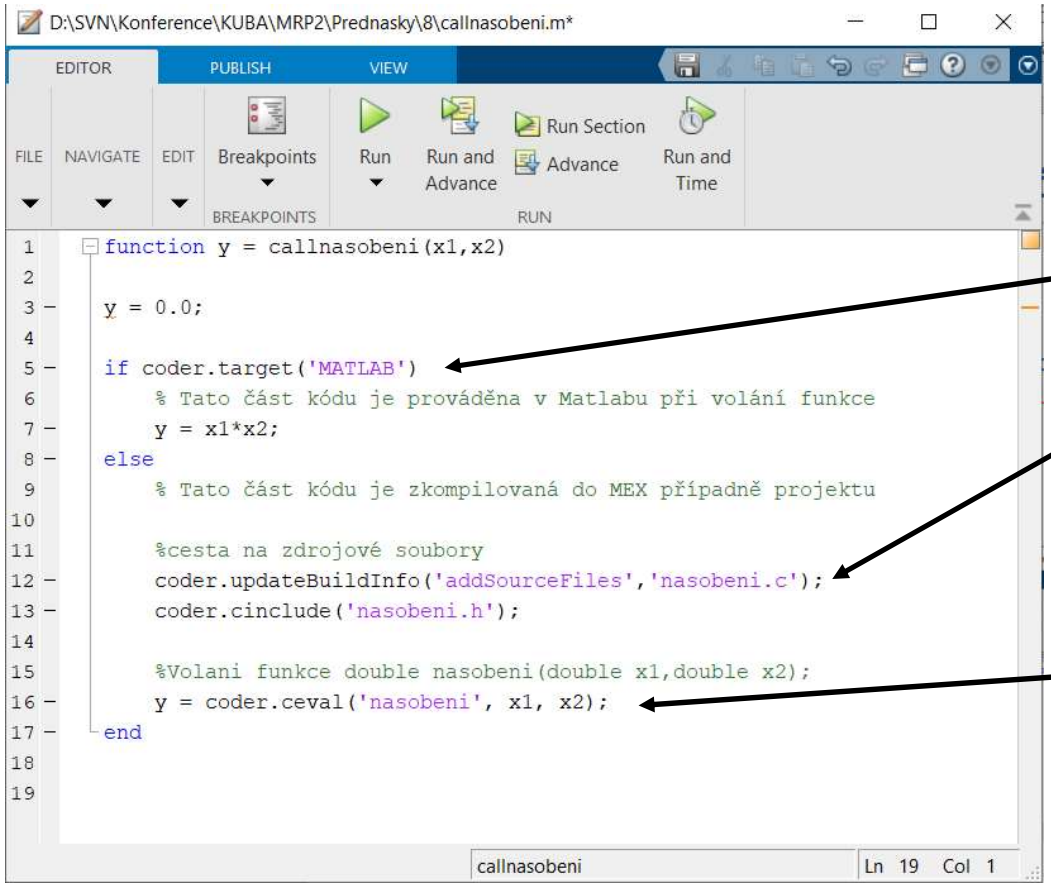
nasobeni.h



Import C funkce do Matlabu

Tvorba MEX z vlastního C pomocí Matlab coder

1) Vytvoříme funkci v Matlabu, která pomocí příkazu coder.ceval volá naší C funkci



Použito pro otestování naší uživatelské funkce

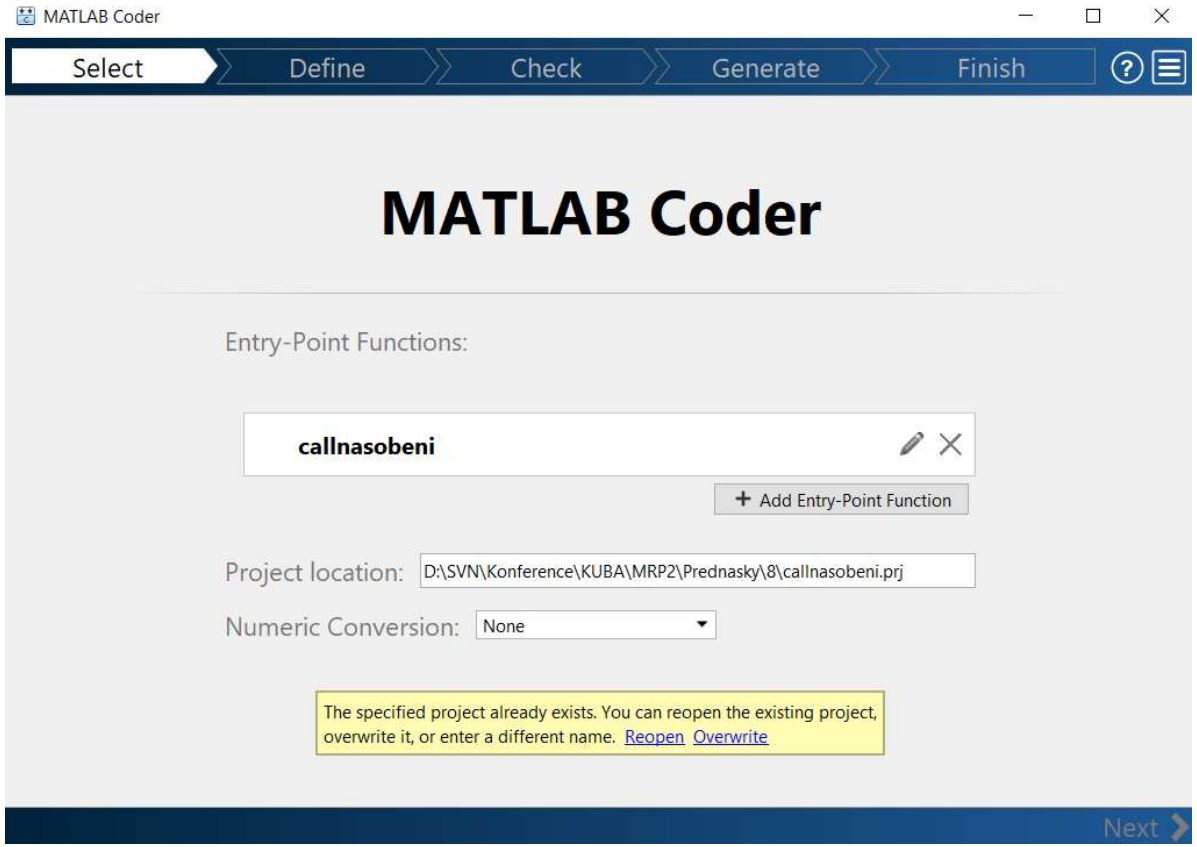
Import uživatelských souborů

Volání uživatelské funkce o dvou proměnných (po zkompilování do mex)

Import C funkce do Matlabu

Tvorba MEX z vlastního C pomocí Matlab coder

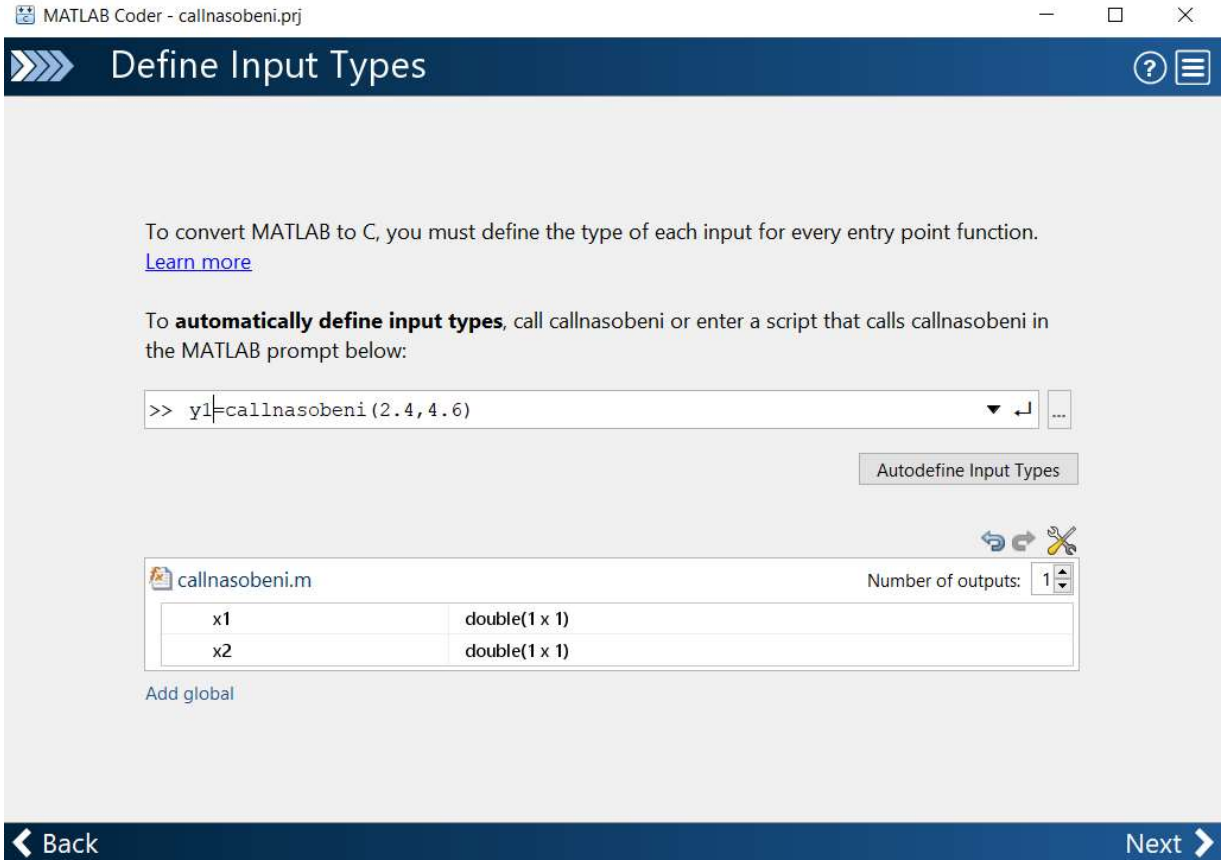
2) Zapneme Matlab coder a vybereme funkci callnasobeni



Import C funkce do Matlabu

Tvorba MEX z vlastního C pomocí Matlab coder

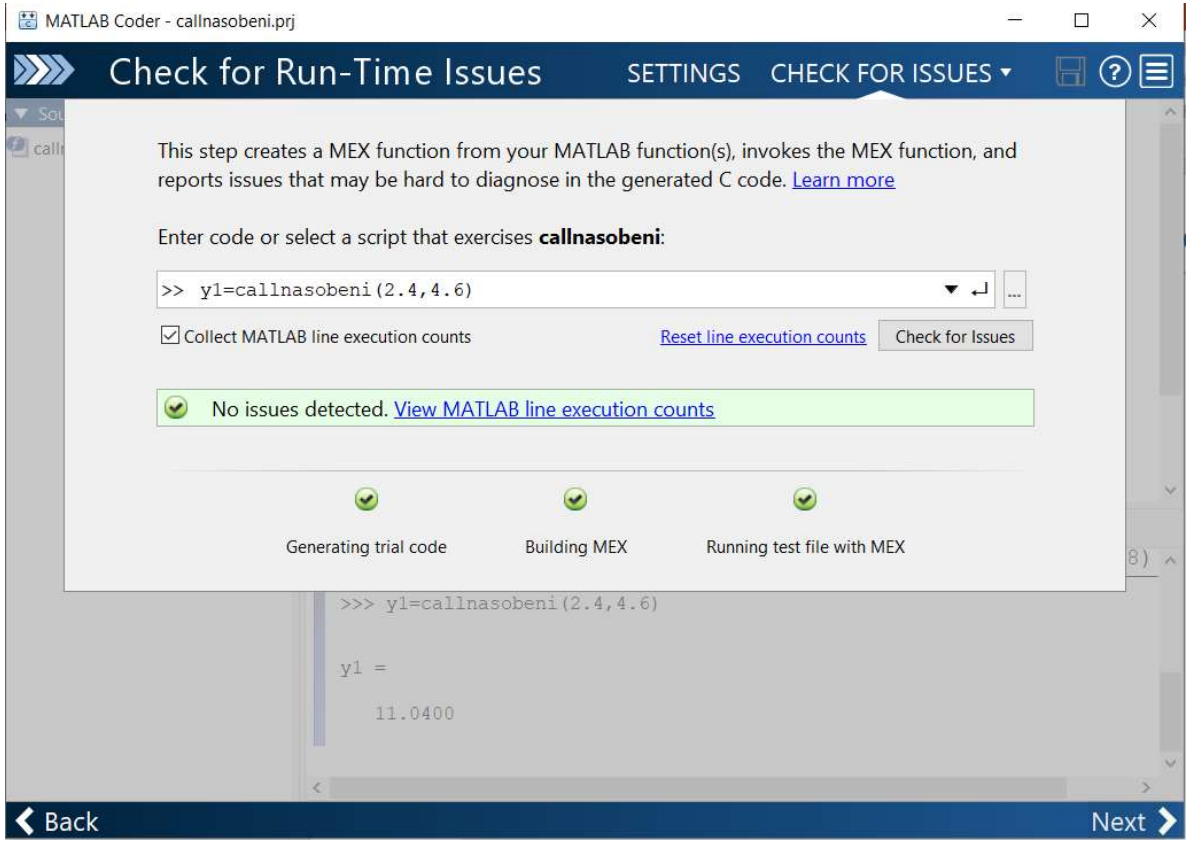
3) Nadefinujeme vstupy funkce callnasobeni



Import C funkce do Matlabu

Tvorba MEX z vlastního C pomocí Matlab coder

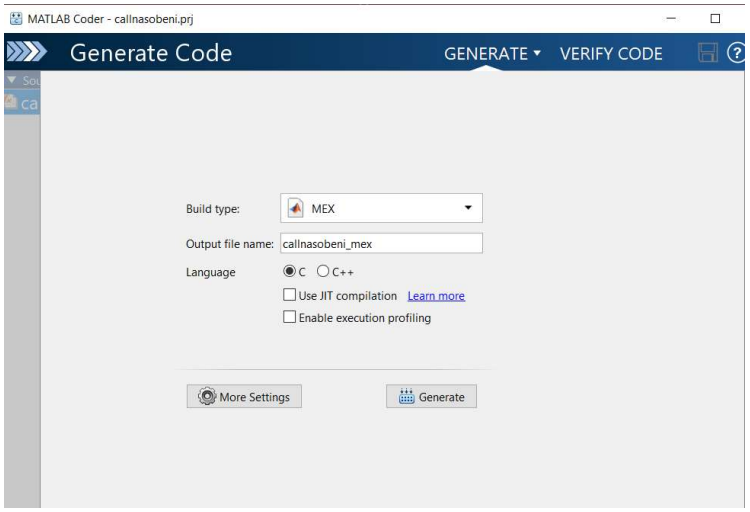
4) Provedeme kontrolu kódu – dojde k porovnání výsledku volané funkce z matlabu a vygenerovaného a zkompilovaného mex souboru

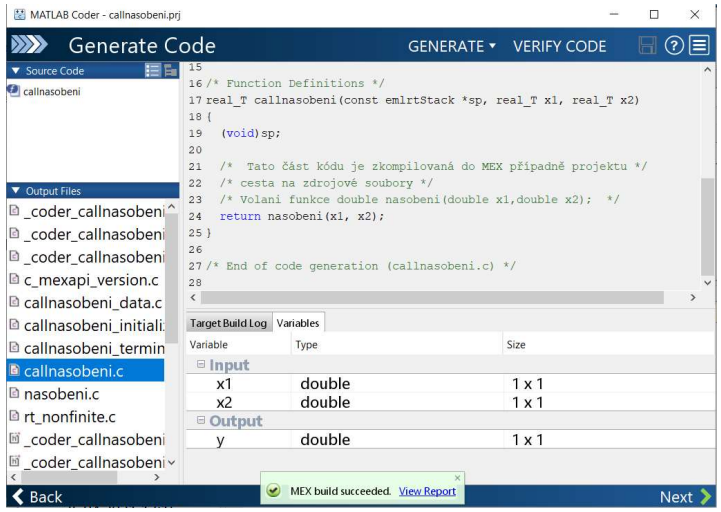


Import C funkce do Matlabu

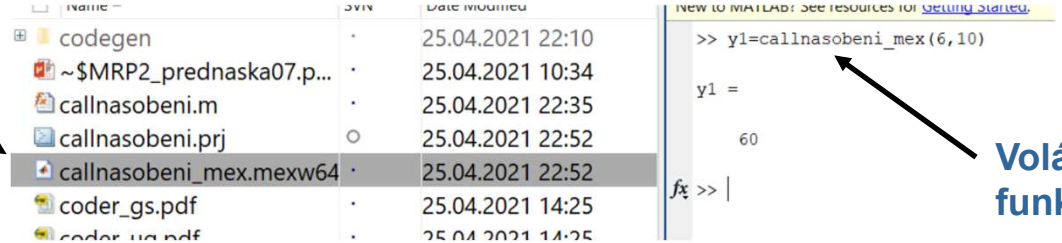
Tvorba MEX z vlastního C pomocí Matlab coder

5) Vygenerujeme mex soubor (znova) a můžeme funkci volat





Vygenerovaný mex



Volání mex (vstupní) funkce

FAKULTA ELEKTROTECHNICKÁ ZÁPADOČESKÉ UNIVERZITY V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Příští přednáška

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