

DR. BOB DAVIDOV

Программный интерфейс MEX. Расширение MATLAB на языке АДА

Цель работы: освоение правил использования кода АДА в среде Simulink.

Задача работы: построение simulink модели с программными блоками языка АДА.

Приборы и принадлежности: персональный компьютер, среда моделирования МатЛАБ Simulink; компилятор GNAT - свободно распространяемая интегрированная среда программирования GNAT (<http://libre.adacore.com/download/configurations>) предназначена для оптимизации кода и компиляции программ, написанных на языке Ада, а так же на С, С++, Asm, С#, Fortran 90, Java, Pascal, Python.

ОБЩИЕ СВЕДЕНИЯ

Ада – единый язык программирования систем реального времени управления автоматизированными комплексами, созданный специально для обеспечения надежности программного кода. Он построен таким образом, чтобы максимальное количество ошибок обнаруживалось на этапе компиляции. Кроме того, одним из требований при разработке языка была максимально лёгкая читаемость текстов программ, даже в ущерб лёгкости написания. По утверждению сторонников, разработка программного обеспечения на Аде в целом обходится на 60 % дешевле, а разработанная программа имеет в 9 раз меньше дефектов, чем при использовании языка Си.

Интеграция среды разработки МатЛАБ с языком SPARK ADA, представляет интерес, например, для синтеза систем управления в МатЛАБ с блоками оригинальных кодов системы написанных на языке АДА.

Взаимодействие МатЛАБ и АДА программ выполняется при помощи компьютерного языка S-функций описания блоков среды Simulink. Помимо языка Ада блоки могут быть написаны на языках С, С++, М (МатЛАБ) и Fortran. S-функции позволяют добавлять собственные блоки в Simulink модели. S-функции можно использовать в Real-Time Workshop.

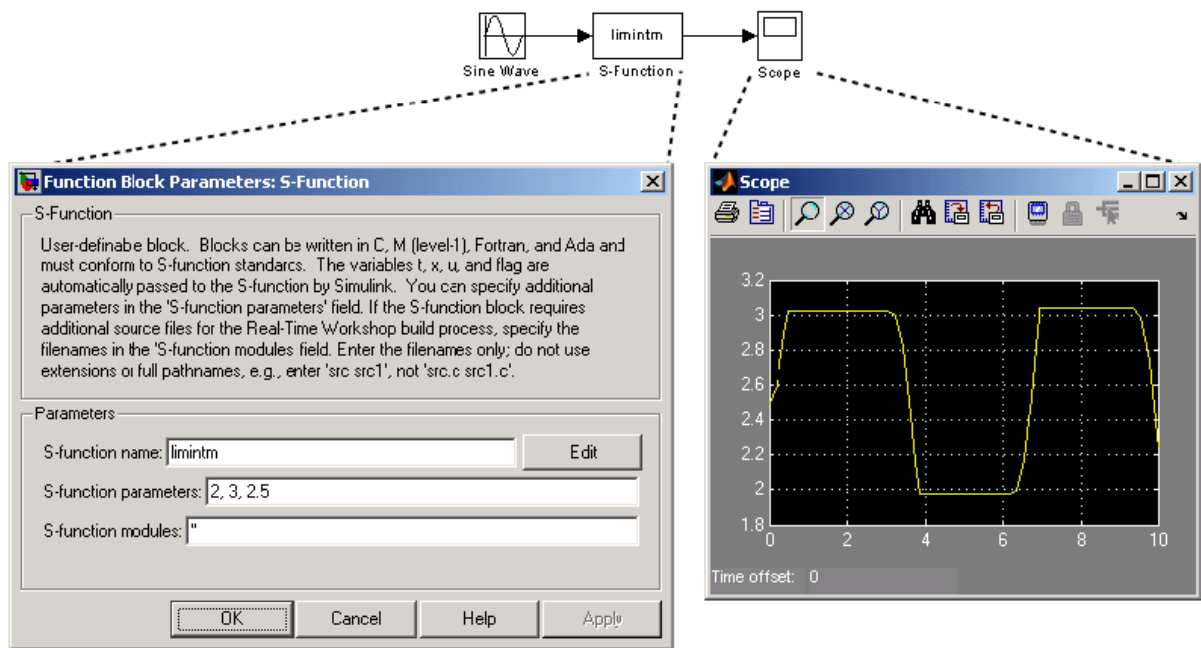


Рис. 1. Поле **S-function parameters** блока S-Function используется для передачи параметров S-функции. Параметры - константы, имена переменных определяемых в рабочем пространстве workspace МатЛАБ или модели. Параметры разделяются запятыми. В этом примере используется три параметра функции `../toolbox/simulink/blocks`, которые задают нижнюю границу, верхнюю границу и начальные значения сигнала.

M-FILE S-FUNCTIONS

М-файл S-функция состоит из функции МАТЛАБ имеющей следующий формат:

`[sys,x0,str,ts]=f(t,x,u,flag,p1,p2,...),`

где `f` - имя S-функции, `x` - вектор состояния соответствующего блока S-функция, `u` - входы блока, `flag` указывает, что задачи должны быть выполнены; `p1`, `p2`, ... - параметры блока. Во время моделирования Simulink неоднократно вызывает `f`, используя флаг, указывающий что задачи должны быть выполнены для конкретного вызова. Каждый раз, когда S-функция выполняет задачу, она возвращает результат в структуру, имеющую формат как показано выше.

Формат S-функции представлен в следующей таблице.

Состояние моделирования	Текущая S-функция	Флаг
Инициализация	<code>mdlInitializeSizes</code>	<code>flag = 0</code>
Вычисление следующего такта	<code>mdlGetTimeOfNextVarHit</code>	<code>flag = 4</code>
Вычисление выходов	<code>mdlOutputs</code>	<code>flag = 3</code>
Обновление дискретных состояний	<code>mdlUpdate</code>	<code>flag = 2</code>
Вычисление производных	<code>mdlDerivatives</code>	<code>flag = 1</code>
Конец задачи моделирования	<code>mdlTerminate</code>	<code>flag = 9</code>

MEX-FILE S-FUNCTIONS

Подобно М-файлу S-функции, MEX-файл функция состоит из набора вызовов процедур для выполнения различных блоков задач. MEX-функция должна соответствовать требованиям Simulink поскольку она вызывает MEX-функцию. Пример MEX-функции на С находится в `matlabroot/simulink/src.sfuntmpl_basic.c`.

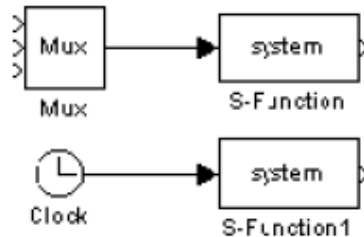


Рис. 2. Примеры подачи вектора (три элемента) и скаляра на вход функции.

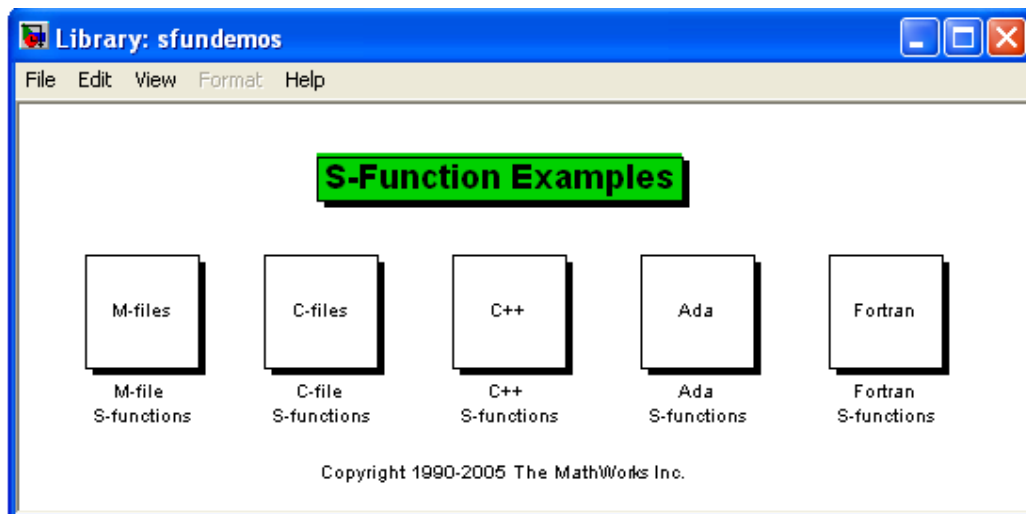


Рис. 3. Примеры S-функций. Simulink Library Browser > User-Defined Functions > S-Function Examples.

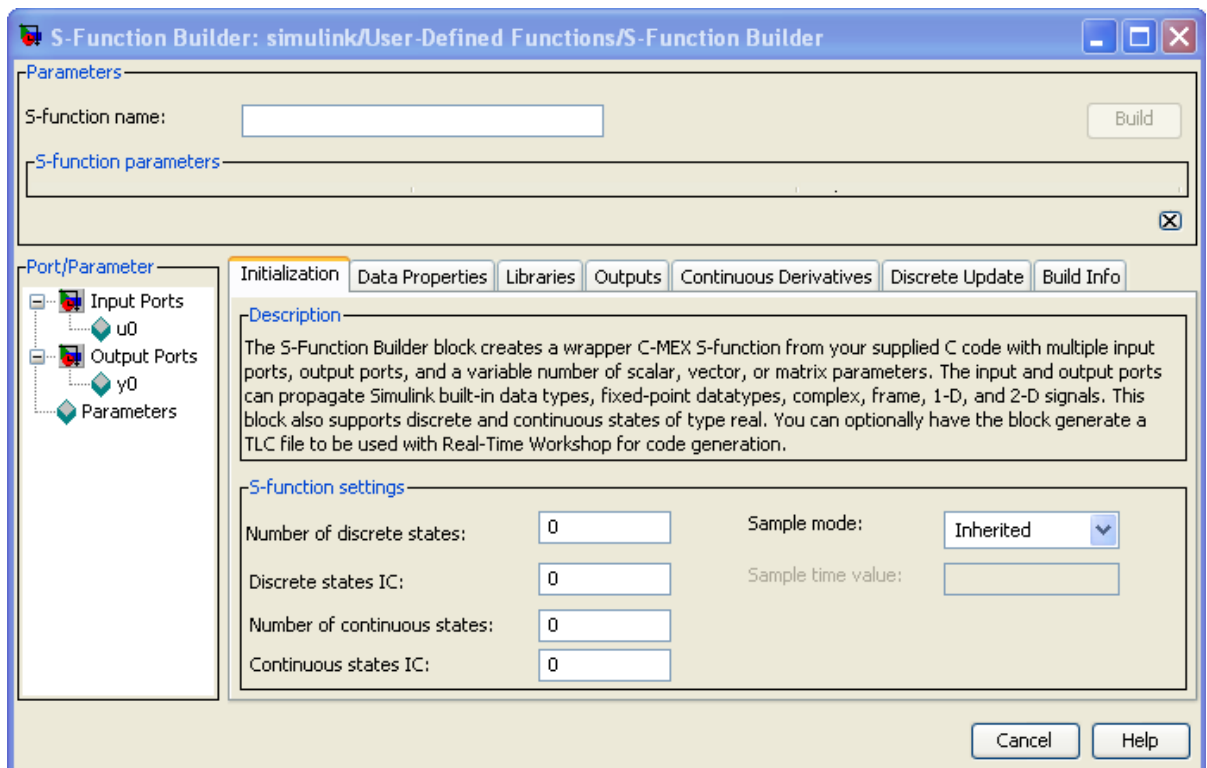
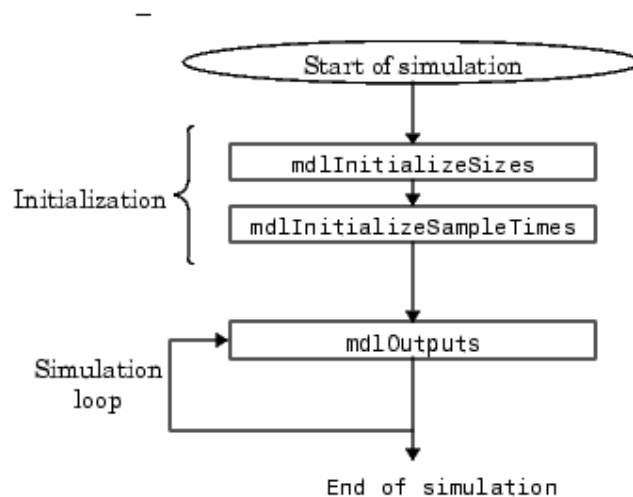


Рис. 4. Построитель S-функций. Simulink Library Browser > User-Defined Functions > S-Function Builder.

S-функция `times_two` используемая в заданиях этой работы содержит методы вызова S-функции, показанные на рисунке.



Код S-функции `times_two` содержит две части:

- Спецификацию пакета (Package Specification)
- Тело пакета (Package body)

Справочный раздел МатЛАБ HELP раскрывает каждую из этих частей.

Вызов **mdlInitializeSizes**(S : in SimStruct) задает количество входных и выходных портов, размеры портов и любые другие объекты (такие как число состояний) необходимых для S-функции. **mdlInitializeSizes** в **times_two** использует **SimStruct** функции входящие в пакет **Simulink** которые задают следующее

- Один входной порт и один выходной port. Ширина входного и выходного портов имеет динамический размер. Это означает, что **Simulink** S-функции может принимать сигнал любой ширины. По умолчанию ширина входного и выходного портов совпадают.
- Одна выборка времени

Метод обеспечивает обработчик исключений для обработки ошибок происходящих при вызове **SimStruct** функций.

Simulink вызывает **mdlOutputs** на каждом временном шаге для расчета выхода блока. Реализация **times_two** метода **mdlOutputs** принимает входной сигнал, умножает его на 2 и записывает результат на выход.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Построение ада S-функции

Для использования S-функций АДА в среде **Simulink** необходимо построить МатЛАБ исполняемый (MEX) файл из программного кода Ada. Для этого используйте **mex** команду:

```
mex [-v] [-g] -ada SFCN.ads,
```

где **SFCN.ads** – имя пакета S-функции; **g** – создает отлаживаемый MEX-файл; **v** – обязывает **Simulink** печатать каждый шаг компиляции во время построения исполняемого файла.

Примечание:

- A) **mex** функция МатЛАБ 7 (R14) компилирует АДА скрипт в бинарный файл с расширением **DLL**
- B) **mex** функция МатЛАБ 2007а компилирует АДА скрипт в бинарный файл с расширением **MEXW32**
- C) **mex** функция МатЛАБ 2012 не компилирует АДА скрипт. В библиотеке **Simulink** блок S-Function не содержит информации о назначении его для АДА программ.

Чтобы построить пример S-функции **times_two** которая должна увеличивать вдвое входной сигнал в среде **Simulink** выполните следующую последовательность.

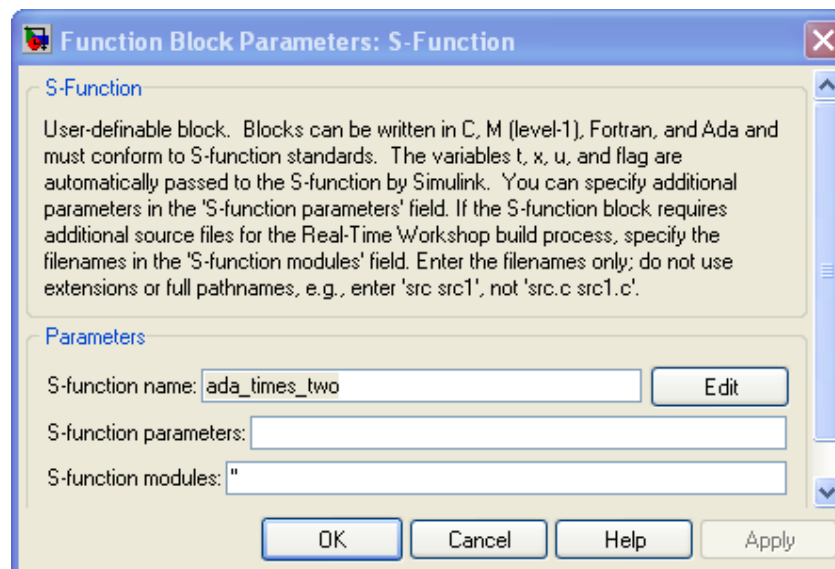
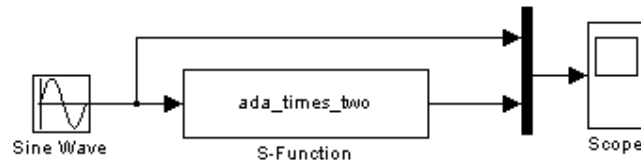
1. Установите компилятор GNAT Ada95 (<http://www.gnu.org/s/gnat/>) версия 3.12 (или более поздняя), например **spark-gpl-2011-i686-pc-mingw32.exe**. Установка создает 'gnatdll' которая необходима для построения MEX интерфейса
2. После установки компилятора перезагрузите компьютер иначе МатЛАБ не “увидит” 'gnatdll'.

3. Скопируйте файлы adb и ads из каталога `..\Program Files\MATLAB\R2007a\simulink\ada\examples\times_two\` в свой рабочий каталог.
4. Наберите в командном окне МатЛАБ:

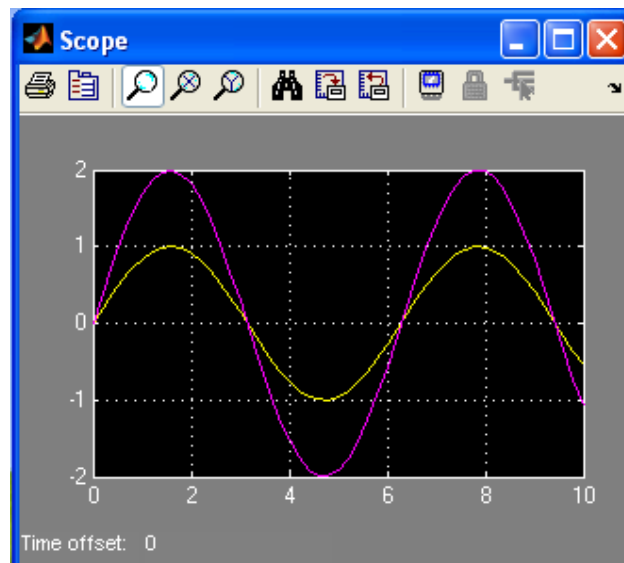
```
>> mex -ada times_two.ads
```

В результате должен быть создан MEX файл `ada_times_two.dll`.

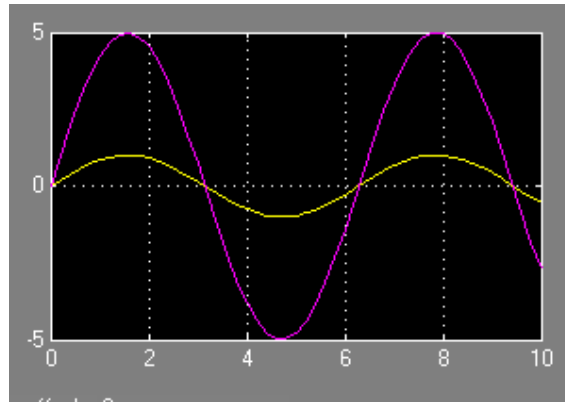
5. В среде Simulink постройте модель



6. Проверьте работоспособность модели по виду входного воздействия и реакции.



7. Найдите способ увеличения входного сигнала модели в пять раз вместо 2-х как показано на рисунке.



Совет: Можно использовать более подробную информацию компиляции Ада кода:
`mex -v -g -ada times_two.ads`. Для устранения указанной компилятором проблемы следует удалить файл `ada_times_two.mexw32` из рабочего каталога.

ВНИМАНИЕ! После редактирования файла необходимо его перекомпилировать.

Задание 2. Построение оболочки АДА проекта для его работы как S-функция в модели Simulink.

1. Внимательно рассмотрите следующий интерфейс (ABC.ads и ABC.ads) проекта B_Cntrl для шести входных параметров и двух - выходных:

Файл ABC.ads

```
-- The Simulink API for Ada S-Function

with Simulink; use Simulink;

package ABC is

  Current_Depth_m : Float;
  Current_Speed_mps : Float;
  Current_Time_s : Float;

  -- The S_FUNCTION_NAME has to be defined as a constant string. Note that
  -- the name of the S-Function (ada_times_two) is different from the name
  -- of this package (times_two). We do this so that it is easy to identify
  -- this example S-Function in the MATLAB workspace. Normally you would use
  -- the same name for S_FUNCTION_NAME and the package.
  --
  S_FUNCTION_NAME : constant String := "ABC";

  -- Every S-Function is required to have the "mdlInitializeSizes" method.
  -- This method needs to be exported as shown below, with the exported name
  -- being "mdlInitializeSizes".
  --
```

```

procedure mdlInitializeSizes(S : in SimStruct);
pragma Export(C, mdlInitializeSizes, "mdlInitializeSizes");

procedure mdlOutputs(S : in SimStruct; TID : in Integer);
pragma Export(C, mdlOutputs, "mdlOutputs");

end ABC;

-- EOF: ABC.ads

```

Файл ABC.adb

```

with Ada.Exceptions; use Ada.Exceptions;
with B_Cntrl;
with Physical_Units;
-- with Math; use Math;

package body ABC is

  -- Function: mdlInitializeSizes -----
  -- Abstract:
  --   Setup the input and output port attributes for this S-Function.
  --
  procedure mdlInitializeSizes(S : in SimStruct) is
    NumInputPorts : Integer := 6; -- ssGetParameterWidth(S,0);

  begin
    -- Set the input port attributes
    ssSetNumInputPorts(S, NumInputPorts);
    for I in 0 .. NumInputPorts-1 loop
      ssSetInputPortWidth(S, I, DYNAMICALLY_SIZED);
      ssSetInputPortDataType(S, I, SS_DOUBLE);
      ssSetInputPortDirectFeedThrough(S, I, TRUE);
      ssSetInputPortOverWritable(S, I, FALSE);
      ssSetInputPortOptimizationLevel(S, I, 3);
    end loop;

    -- Set the output port attributes
    ssSetNumOutputPorts(S, 2);

    ssSetOutputPortWidth(S, 0, DYNAMICALLY_SIZED);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);
    ssSetOutputPortOptimizationLevel(S, 0, 3);

    ssSetOutputPortWidth(S, 1, DYNAMICALLY_SIZED);
    ssSetOutputPortDataType(S, 1, SS_DOUBLE);
    ssSetOutputPortOptimizationLevel(S, 1, 3);

    -- Set the block sample time.
    ssSetSampleTime(S, INHERITED_SAMPLE_TIME);
  -- ssSetSampleTime(S, INHERITED_SAMPLE_TIME, 0.0);

```

```

exception
  when E : others =>
    if ssGetErrorStatus(S) = "" then
      ssSetErrorStatus(S,
        "Exception occured in mdlInitializeSizes. " &
        "Name: " & Exception_Name(E) & ", " &
        "Message: " & Exception_Message(E) & " and " &
        "Information: " & Exception_Information(E));
    end if;
end mdlInitializeSizes;

-- Function: mdlOutputs -----
--   Compute the S-Function's output, given its input
procedure mdlOutputs(S : in SimStruct; TID : in Integer) is

  use B_Cntrl;
  use Physical_Units;

  -- Input Ports:
  uWidth : Integer := ssGetInputPortWidth(S,0);
  In_1 : array(0 .. uWidth-1) of Real_T; -- Real_T: SS_DOUBLE (simulink); real32_T:
SS_SINGLE (simulink) ..
  for In_1'Address use ssGetInputPortSignalAddress(S,0);
  In_2 : array(0 .. uWidth-1) of Real_T;
  for In_2'Address use ssGetInputPortSignalAddress(S,1);
  In_3 : array(0 .. uWidth-1) of Real_T;
  for In_3'Address use ssGetInputPortSignalAddress(S,2);
  In_4 : array(0 .. uWidth-1) of Real_T;
  for In_4'Address use ssGetInputPortSignalAddress(S,3);
  In_5 : array(0 .. uWidth-1) of Real_T;
  for In_5'Address use ssGetInputPortSignalAddress(S,4);
  In_6 : array(0 .. uWidth-1) of Real_T;
  for In_6'Address use ssGetInputPortSignalAddress(S,5);

  -- Input Params:
  Target_Depth    : Float;
  Max_Descent_Rate : Float;
  Max_Ascent_Rate  : Float;

  -- Output Ports:
  yWidth : Integer := ssGetOutputPortWidth(S,0);
  Out_1   : array(0 .. yWidth-1) of Real_T; -- boolean_T: SS_BOOLEAN (simulink);
  for Out_1'Address use ssGetOutputPortSignalAddress(S,0);
  Out_2   : array(0 .. yWidth-1) of Real_T;
  for Out_2'Address use ssGetOutputPortSignalAddress(S,1);

  -- Output params of Go_To_Depth.adb;
  ABC_state : ABC_Operations;

begin
  Target_Depth := Float(In_1(0));

```

```

Max_Descent_Rate := Float (In_2(0));
Max_Ascent_Rate  := Float (In_3(0));
Current_Depth_m  := Float (In_4(0));
Current_Speed_mps:= Float (In_5(0));
Current_Time_s   := Float (In_6(0));

-- VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
-- Start of B_Cntrl_Output.Go_To_Depth

      ABC_state := B_Cntrl.Go_To_Depth (Physical_Units.metres (Target_Depth),
m_per_s (Max_Descent_Rate), m_per_s (Max_Ascent_Rate));

-- End of B_Cntrl_Output.Go_To_Depth
-- VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV

if Valve.Inj_Valve_On_Off_Bit then
    Out_1(0) := 1.0;
else
    Out_1(0) := 0.0;
end if;

if Valve.Drain_Valve_On_Off_Bit then
    Out_2(0) := 1.0;
else
    Out_2(0) := 0.0;
end if;

exception
when E : others =>
    if ssGetErrorStatus(S) = "" then
        ssSetErrorStatus(S,
            "Exception occurred in mdlOutputs. " &
            "Name: " & Exception_Name(E) & ", " &
            "Message: " & Exception_Message(E) & " and " &
            "Information: " & Exception_Information(E));
    end if;
end mdlOutputs;

end ABC;

-- EOF: ABC.adb
```

2. Разместите файлы оболочки и проекта (прилагаются к этой работе) в рабочей директории МатЛАБ

Name	↑Ext	Size
↑ [..]		<DIR>
ABC	adb	5 663
ARM_Hardware_Emulator	adb	4 469
B_Cntrl	adb	18 400
Config	adb	1 191
Math	adb	6 943
Physical_Units	adb	1 691
ABC	ads	1 898
ARM_Hardware_Emulator	ads	1 538
B_Cntrl	ads	2 751
Config	ads	1 007
Generic_Types	ads	1 215
Math	ads	2 538
Physical_Units	ads	3 191

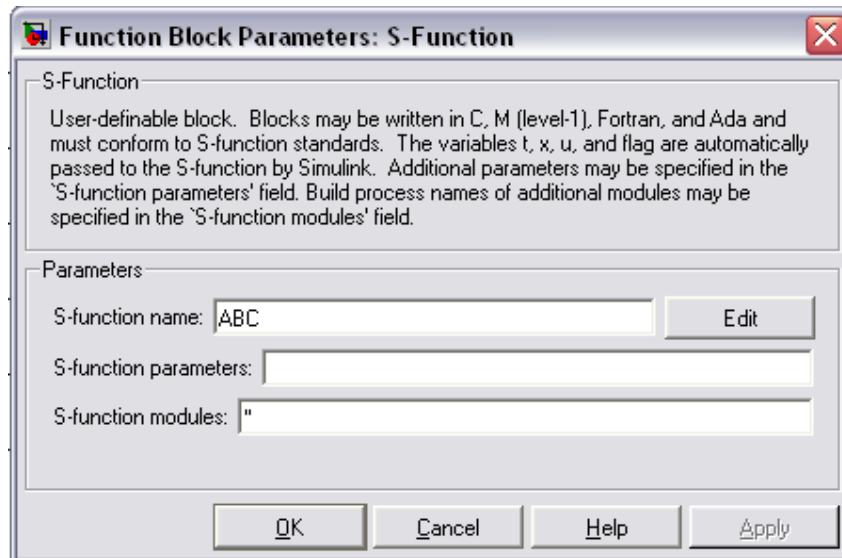
3. Откомпилируйте ABC проект.

>> `mex -ada ABC.ads`

В результате должен получиться файл динамической библиотеки `ABC.dll`

Name	↑Ext	Size
↑ [..]		<DIR>
↑ [ABC_ada_sfcn_win32]		<DIR>
ABC	adb	5 663
ARM_Hardware_Emulator	adb	4 469
B_Cntrl	adb	18 400
Config	adb	1 191
Math	adb	6 943
Physical_Units	adb	1 691
ABC	ads	1 898
ARM_Hardware_Emulator	ads	1 538
B_Cntrl	ads	2 751
Config	ads	1 007
Generic_Types	ads	1 215
Math	ads	2 538
Physical_Units	ads	3 191
ABC	dll	426 461

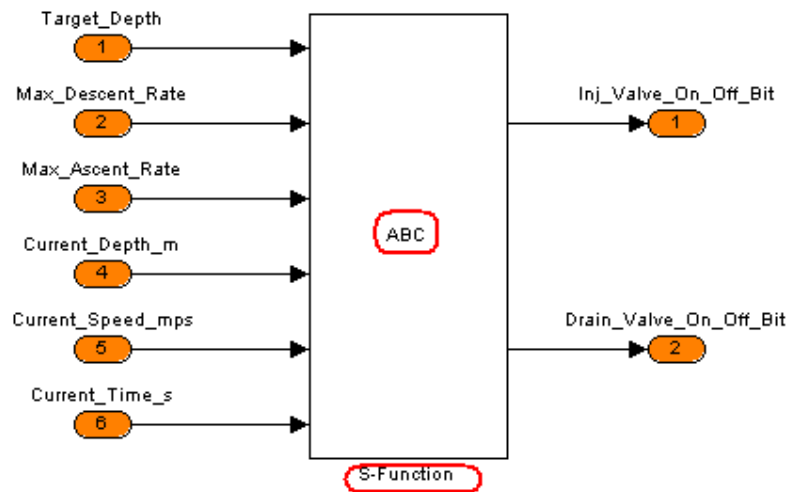
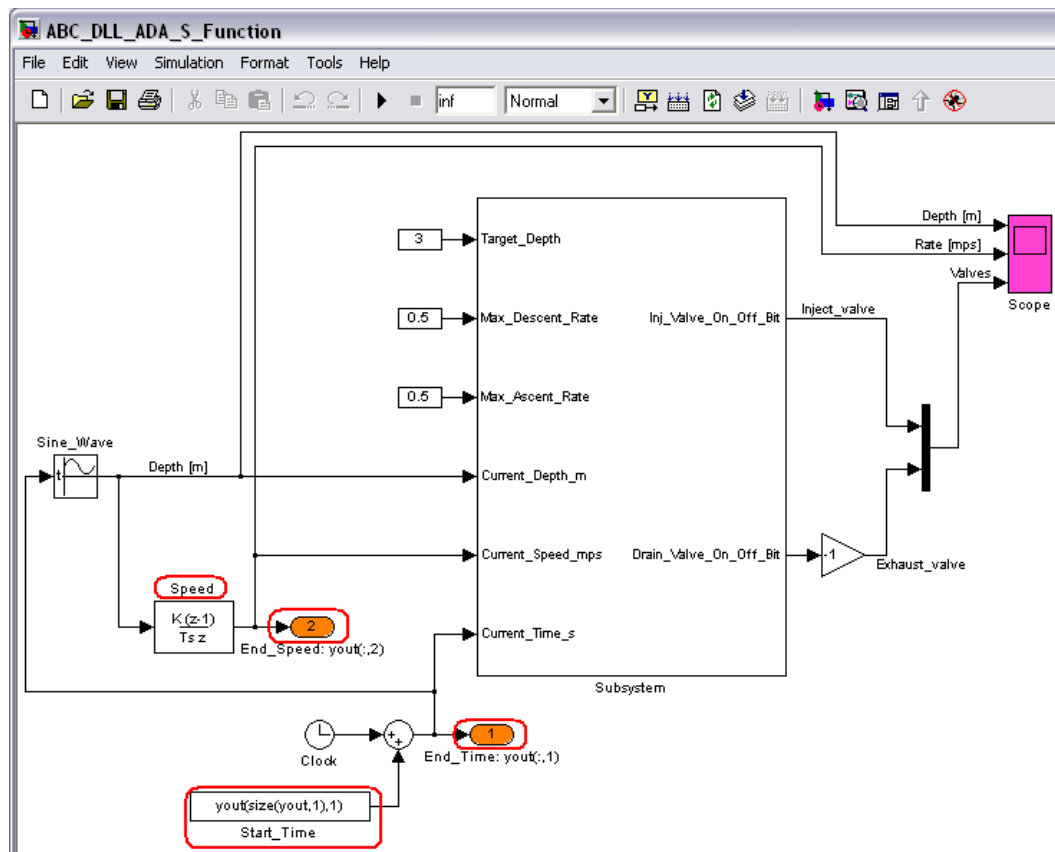
4. Подключите **ABC.dll** к Simulink модели, например, **ABC_DLL_ADA_S_Function.mdl**, через S-function блок

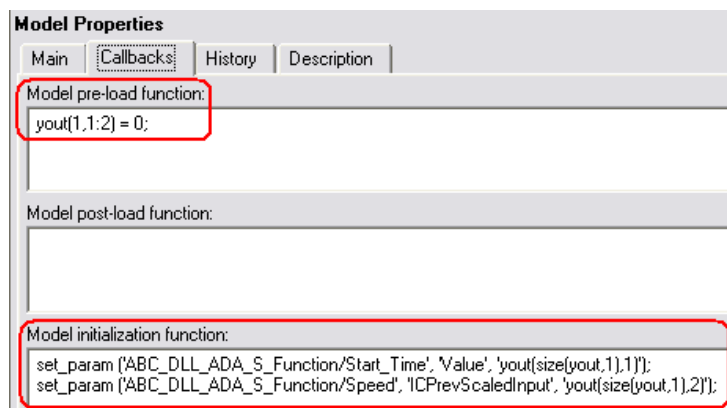
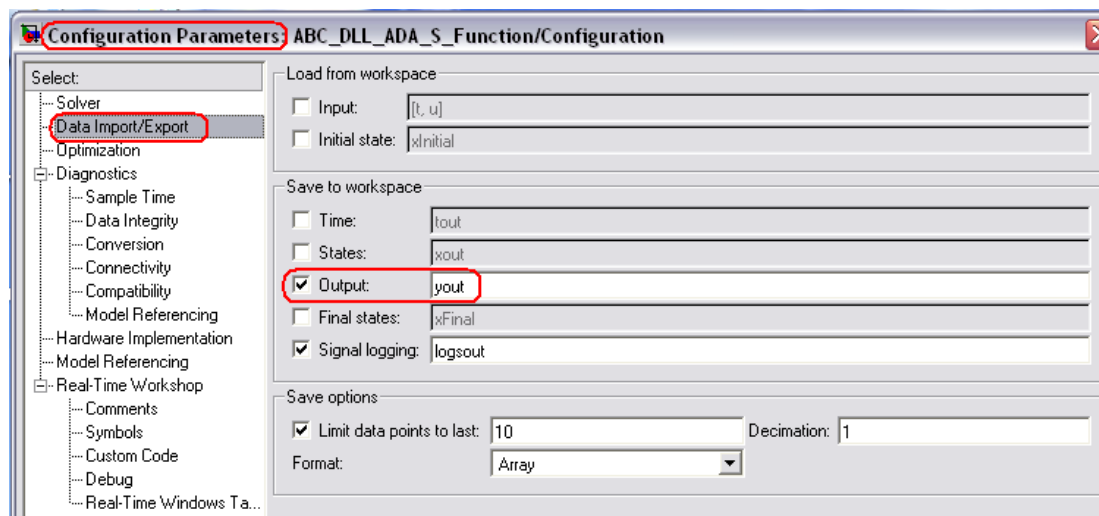


Внимание: После каждого редактирования АДА файлов и последующей перекompиляции необходимо очистить окно блока S-function \ "S-function name" и вновь ввести имя DLL функции.

5. Запустите модель с АДА S-функцией на выполнение. Убедитесь в работоспособности Simulink модели с интегрированным модулем языка АДА.

Внимание: Если при перезапуске модели АДА S-функция требует конечные значения предыдущего этапа моделирования как начальные значения нового этапа моделирования, необходимо это обеспечить, например, как показано ниже.





КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как обеспечить совместную работу модели Simulink с модулем на языке АДА в режиме реального времени?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Help МатЛАБ
2. А. Гавва. Адское" программирование. Ada-95. Компилятор GNAT, V-0.4, - 2004,