Simulink S-Function for RT-LAB

Document 1b

Creation of a S-Function From C Code and Protection of the Source Code

Version 1.2

www.opal-rt.com

1751 Richardson, suite 2525
Montréal (Québec) Canada H3K 1G6

www.opal-rt.com

Contents

REVISION HISTORY

Version	Date	Responsible	Comment
1.0	2017-02-20	Daniel O'Brien	Initial version
1.1	2017-03-02	Daniel O'Brien	Minor improvements.
1.2	2017-09-06	Daniel O'Brien	• Add references toward the procedure for configuring a virtual machine. • Add information on "gmake: *** No rule to make target `–f'. Stop." error.

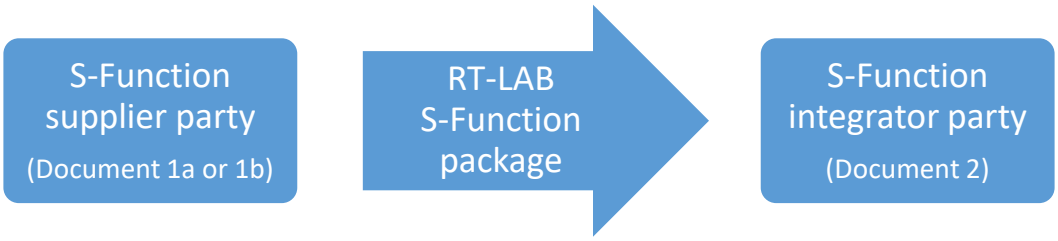
INTRODUCTION

This document presents a procedure to:

- Create a S-Function compatible with RT-LAB from:
  - C code already prepared for a S-Function interface
- Compile the source code of the S-Function in order to hide intellectual property

The result of this procedure is a package with everything needed for another party to integrate your S-Function in a model designed for real-time simulation in RT-LAB. For this purpose, there is a second document:

- Document 2 – Integration of an Existing S-Function in a Model for RT-LAB



In this procedure, the user is expected to know how to implement a S-Function with C code for a Simulink model. If this is not the case, MathWorks provides information on getting started:

- https://www.mathworks.com/help/simulink/s-function-basics.html

REQUIRED MATERIAL

- Windows host PC
- Target:

<table>
<tr><th></th><th>OPAL-RT Linux (x86-based)
(previously known as RedHat)</th><th>Windows
(localhost target in RT-LAB)</th></tr>
<tr><td rowspan="2">Requirements</td><td>OPAL-RT simulator
(OP5600, OP4510, …)
or
Vitual machine with
OPAL-RT Linux
(x86-based)***</td><td rowspan="2">Visual C++ compiler from Visual Studio 2010
You may install
Microsoft Visual Studio 2010
or
Microsoft Visual C++ 2010
Express (free).</td></tr>
<tr><td>MobaXterm software
(for transferring files on
the target)</td></tr>
</table>

- MATLAB/Simulink
    - The same version (ex: R2013b) and architecture (ex: 64-bit) of MATLAB/Simulink must be used by the supplier and the integrator of the S-Function.

*** Procedure for configuring a virtual machine:
https://www.opal-rt.com/KMP/index.php?/article/AA-01281/8/HowTo/How-to-create-a-Virtual-machine-with-OPAL-RT-Linux.html

CONSTRAINTS FOR REAL-TIME S-FUNCTIONS

RT-LAB is compatible with the following type of S-Function:

- Level-2 C S-Function (C MEX S-Function)

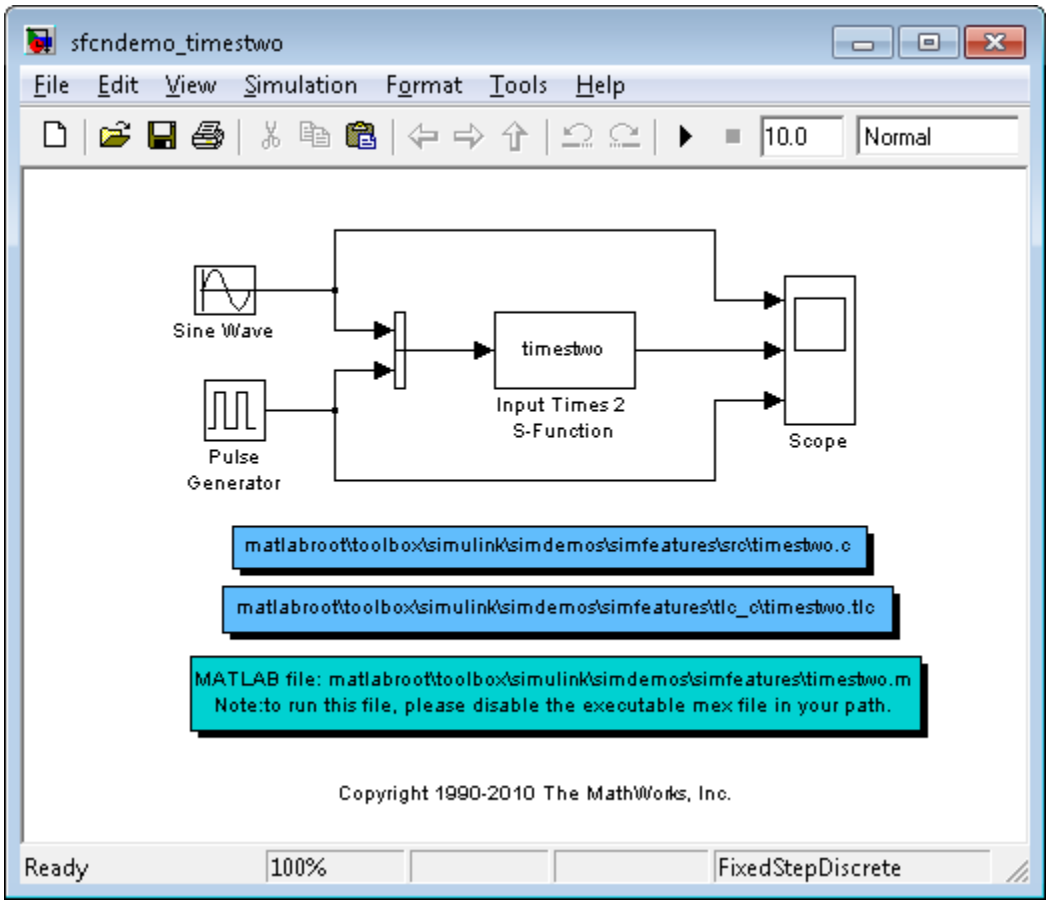
This means the S-Function must be implemented in C code. You must also have access to the source code, and not only a MEX file. Having access to the source code is necessary in order create a binary that will be compatible with RT-LAB's software environment (Linux kernel version, compiler, C library, etc.).

Furthermore, the S-Function shall adhere to the following real-time implementation restrictions:

- Use compatible and standard libraries compatible with one of the compiler on the real-time simulation platform:
  - OPAL-RT Linux (x86-based) targets:
    - GCC 4.1.2 (GNU libc 2.5)
  - Windows targets:
    - Visual C++ 2010 compiler
- No dynamic memory allocation. All memory allocation should be done during initialization of the simulation
- No operating system interrupts
- Avoid using specific semaphores, processes, thread, and synchronization functions during real-time simulation
- Avoid using file access for (e.g. for reading and writing data) during real-time simulation
- Avoid iterative or other computing structures, such as iteratively solving algebraic equations within each time-step or "do … while" loops, where the execution time can vary greatly

SECTION 1: S-FUNCTION CREATION

As an example, a S-Function will be created using the timestwo example provided in an installation of MATLAB/Simulink.



1. In MATLAB, type the following command in order to set a default compiler. You may choose any compiler detected by MATLAB.

```
mex -setup
```

Note: The `mex` command varies from one version of MATLAB to another. Please refer to MathWorks' documentation in order to properly configure a compiler.

2. Change the current folder in MATLAB to the folder where the Simulink model is saved.

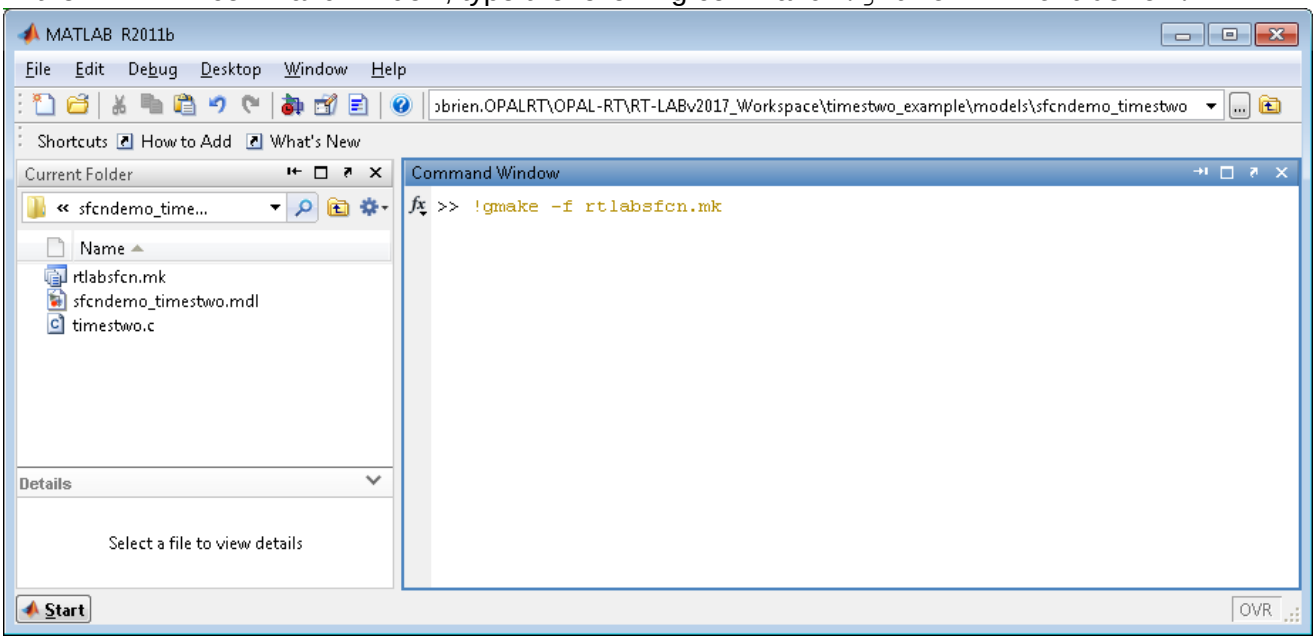
3. Copy the following file that is provided with the procedure next to the Simulink model.
   - rtlabsfcn.mk

4. Make sure the source code of the S-Function is in the same directory. Example: timestwo.c

5. Edit rtlabsfcn.mk in order to configure the user-defined parameters at the beginning of the file.

```
# File     : rtlabsfcn.mk
#
# Abstract:
#       Makefile for building comptible S-Functions for RT-LAB.

# ------------------------------------------------------------------
# USER-DEFINED PARAMETERS
# ------------------------------------------------------------------

# Name of the S-Function
# Must also be the name of the primary source file without file extension.
# Must also be same name has the macro in the C file (#define S_FUNCTION_NAME  sfun_controller)
# Example: SFCN_NAME = sfun_controller
SFCN_NAME = timestwo

# Additional source files.
# Leave empty if there are none.
# Separate multiple additional source files with a space.
# Example: ADD_SRCS = additional_src1.c additional_src2.c
ADD_SRCS =

# MATLAB Root.
# Example: MATLAB_ROOT = C:\Program Files (x86)\MATLAB\R2013b
MATLAB_ROOT = C:\Program Files (x86)\MATLAB\R2011b

# ------------------------------------------------------------------
```

6. In the MATLAB command window, type the following command: `!gmake -f rtlabsfcn.mk`



This will create a MEX file. Even if you are already familiar with the creation of MEX files, it is recommend to use the provided method. This will validate that the modifications made to rtlabsfcn.mk are correct for the code compilation section.

Note: If you receive the error "*gmake: *** No rule to make target `–f'. Stop.*", it may be because of a bad character conversion during the copy-paste between the current document and the *MATLAB Command Window*. Please type the command manually to solve the error.

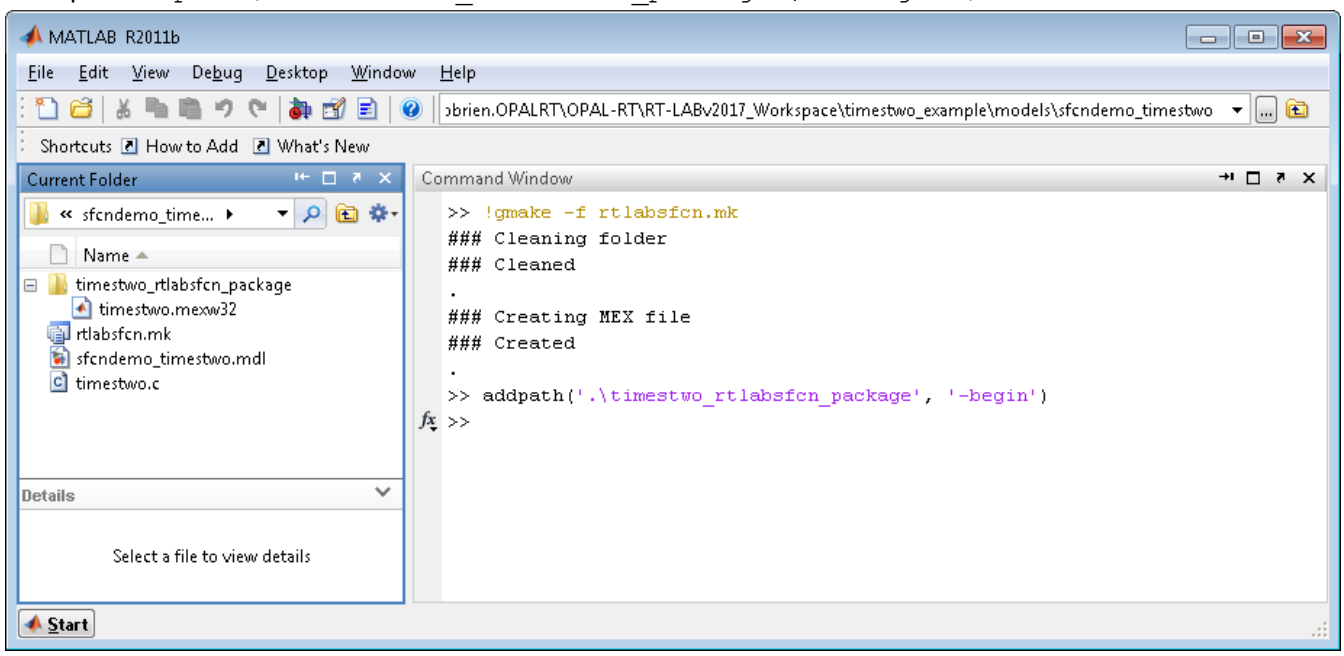
Note: It is also possible to create a MEX file using the *mex* MATLAB command. However, the current step in the procedure will help validate the information you entered in *rtlabsfcn.mk*. This will prevent errors that are harder to debug during the code compilation on the target platform.

7. Make sure you see the following message at the end of the execution of the script.

```
### Creating MEX file
### Created
```

8. Add the path of the RT-LAB S-Function package folder that has just been created to MATLAB's search paths.

   Example: `addpath('.\timestwo_rtlabsfcn_package', '-begin')`



9. Make sure that the model runs in Simulink without any problem at this point. In other words, you must be able to press *Start simulation* in Simulink.

SECTION 2: CODE COMPILATION (FOR HIDDING THE SOURCE CODE)

Now that the S-Function is generated, its source code must be compiled in order to hide any intellectual property. The procedure is different depending on the platform of the target you will use with RT-LAB.

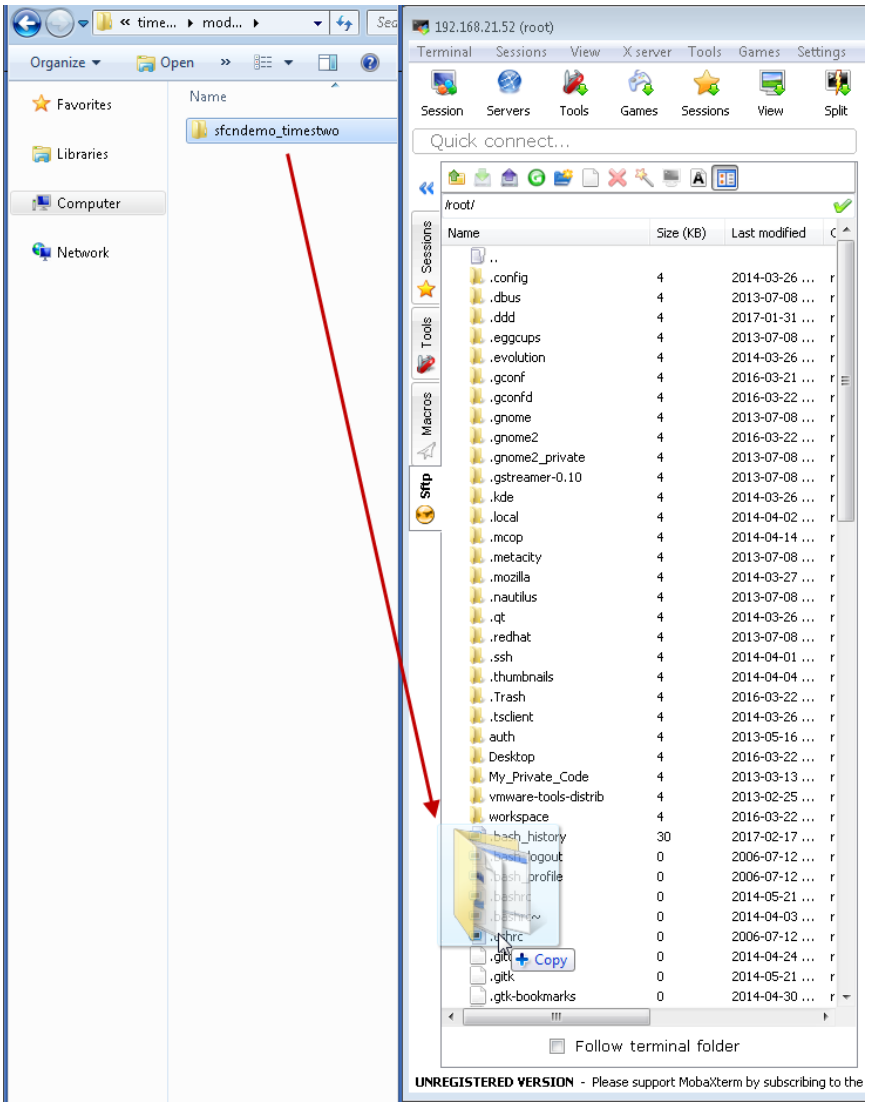
Note: Throughout this section, *timestwo* is used as the name of the S-Function. You may replace it by the name of your S-Function.

OPAL-RT LINUX (X86-BASED) TARGET

For *OPAL-RT Linux* target platforms, you must either have access to an OPAL-RT simulator, or configure a virtual machine by following the procedure described here:
https://www.opal-rt.com/KMP/index.php?/article/AA-01281/8/HowTo/How-to-create-a-Virtual-machine-with-OPAL-RT-Linux.html

1. Follow Annex 1 in order to install and configure MobaXterm. You may use an alternative ssh/sftp client if you are more familiar with it.

2. Drag and drop the folder containing rtlabsfcn.mk and the source code to MobaXterm's *sftp* tab.



3. In the terminal, write the following commands:

```
cd /root/sfcndemo_timestwo
make -f rtlabsfcn.mk build_linux_x86
```

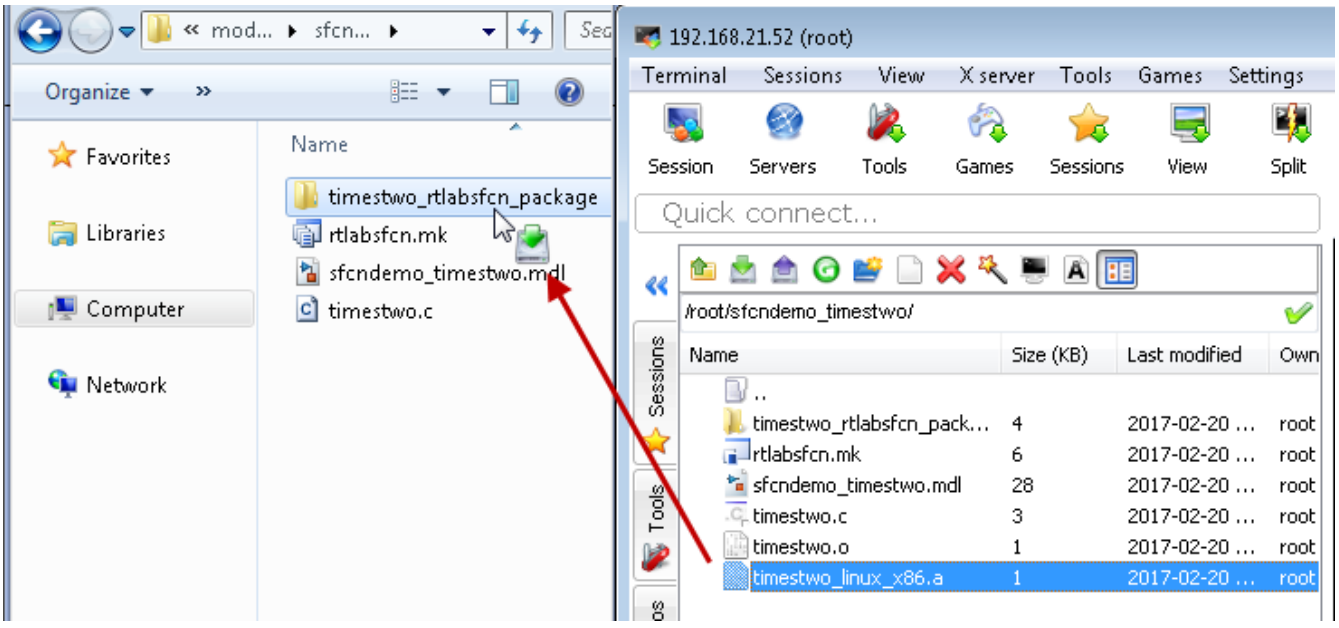
Note: *sfcndemo_timestwo* is the name of the folder that contains rtlabsfcn.mk and the source code.

4. Make sure you see the following message at the end of the execution of the script.

```
SUCCESSFUL GENERATION OF LINUX X86 .A LIBRARY
```

```
[root@RTServer sfcndemo_timestwo]# cd /root/sfcndemo_timestwo
[root@RTServer sfcndemo_timestwo]# make -f rtlabsfcn.mk build_linux_x86


.

gcc -c -O2 -ffast-math -mtune=i686 -march=i686 -falign-loops=2 -falign-jumps
=2 -falign-functions=2 -DMODEL=timestwo -DRT -DUSE_RTMODEL -I/usr/matlab/v7.
13/simulink/include -I/usr/matlab/v7.13/extern/include -I/usr/matlab/v7.13/r
tw/c/src -I/usr/matlab/v7.13/rtw/c/src/matrixmath -I/usr/matlab/v7.13/rtw/c/
libsrc -I/usr/matlab/v7.13/toolbox/simscape/include/drive -I/usr/matlab/v7.1
3/toolbox/simscape/include/mech -I/usr/matlab/v7.13/toolbox/simscape/include
/foundation -I/usr/matlab/v7.13/toolbox/simscape/include/network_engine -I/u
sr/matlab/v7.13/toolbox/simscape/include/ne_sli -I/usr/matlab/v7.13/toolbox/
dspblks/include -I/usr/matlab/v7.13/toolbox/simscape/include/compiler/core/c
 -I/usr/matlab/v7.13/toolbox/simscape/include/engine/sli/c -I/usr/matlab/v7.
13/toolbox/simscape/include/engine/core/c -I/usr/matlab/v7.13/toolbox/simsca
pe/include/utils -I.  timestwo.c

.

ar rc timestwo_linux_x86.a timestwo.o
GNU ranlib 2.17.50.0.6-6.el5 20061020
Copyright 2005 Free Software Foundation, Inc.
This program is free software; you may redistribute it under the terms of
the GNU General Public License.  This program has absolutely no warranty.


.
SUCCESSFUL GENERATION OF LINUX X86 .A LIBRARY
.
[root@RTServer sfcndemo_timestwo]#
```

5. In MobaXterm's *sftp* tab, go inside the */root/sfcndemo_timestwo* folder. You should see a file named *timestwo_linux_x86.a* that is the result of the compilation process. Drag and drop it inside the *timestwo_rtlabsfcn_package* folder on the Windows PC.



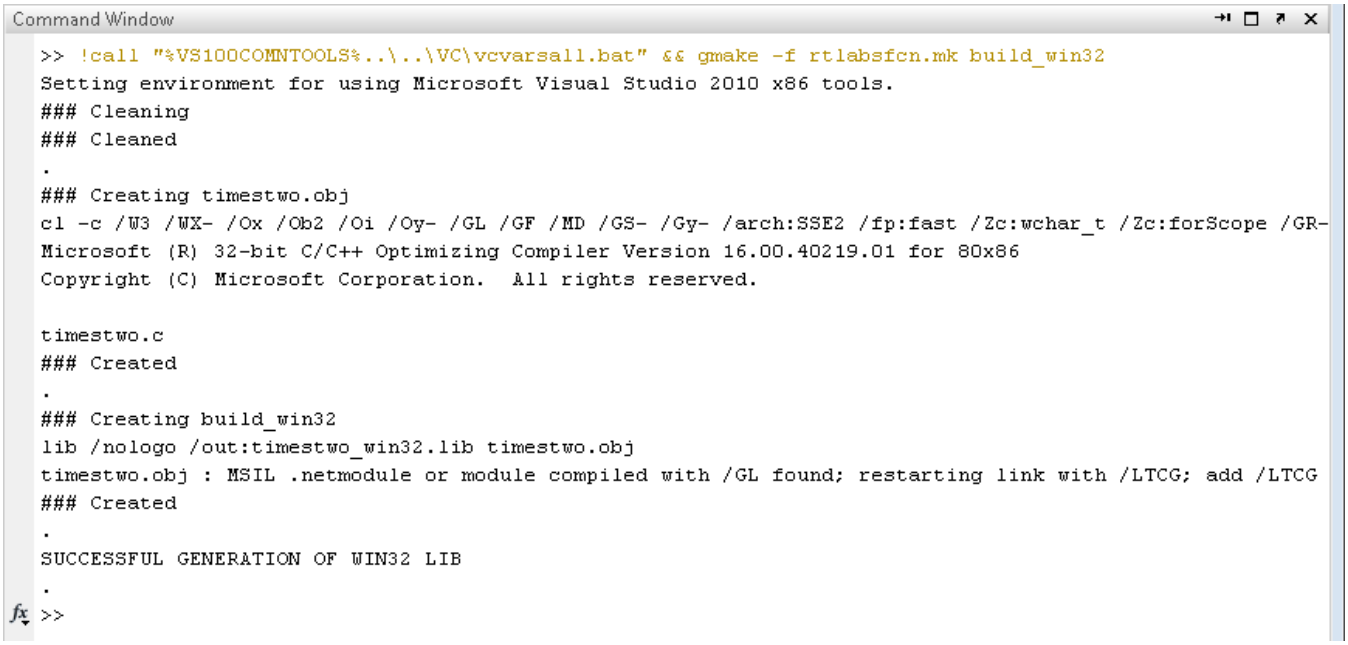
WINDOWS TARGET

1. In order to compile the source code for a Windows target, the Visual C++ 2010 compiler must be installed.

2. Go back to the MATLAB command window, and type the following command:
   `!call "%VS100COMNTOOLS%..\..\VC\vcvarsall.bat" && gmake -f rtlabsfcn.mk build_win32`

3. Make sure you see the following message at the end of the execution of the script.
   `SUCCESSFUL GENERATION OF WIN32 LIB`

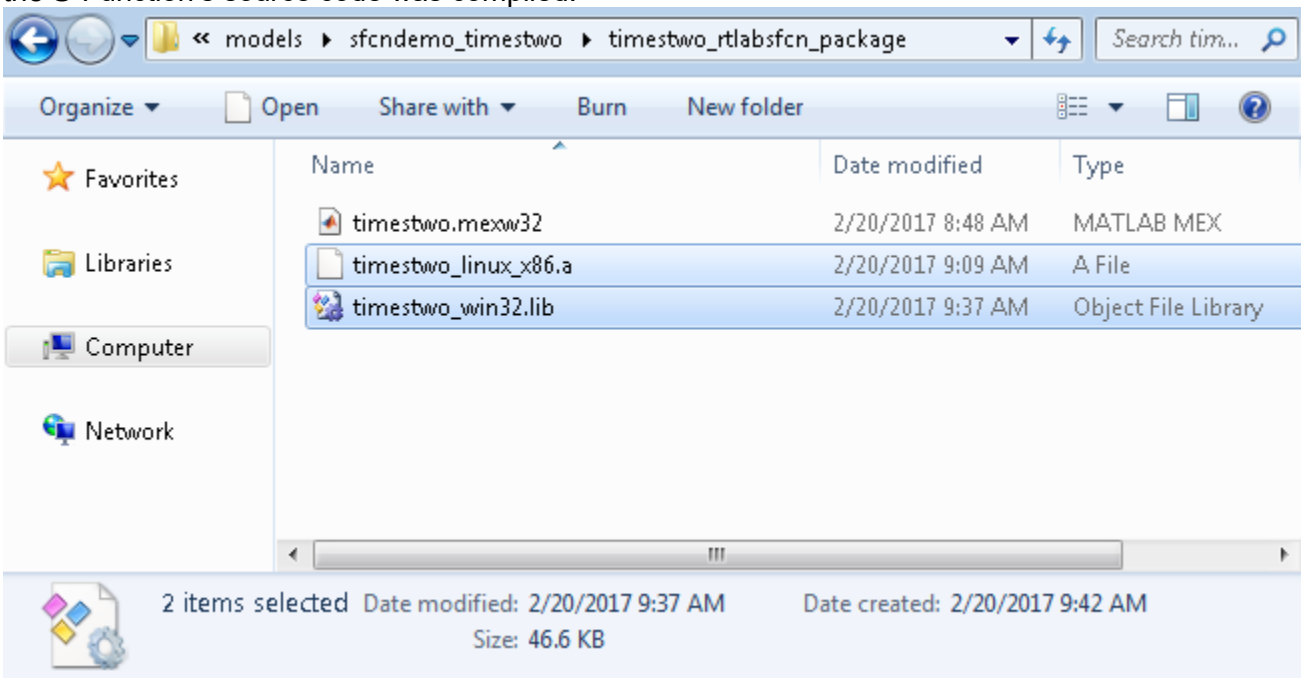
   If you encounter a fatal error that mentions `Cannot open include file: 'rtwtypes.h'`, copy the following file that is provided with the procedure next to the Simulink model.
   - rtwtypes.h



4. You should see a file named *timestwo_win32.lib* that is the result of the compilation process. Copy it inside the *timestwo_rtlabsfcn_package* folder.

RT-LAB S-FUNCTION PACKAGE

The resulting *timestwo_rtlabsfcn_package* should look like the following depending on which target platforms the S-Function's source code was compiled.



- *timestwo_linux_x86.a* is for OPAL-RT Linux (x86-based) targets.
- *timestwo_win32.lib* is for Windows targets.

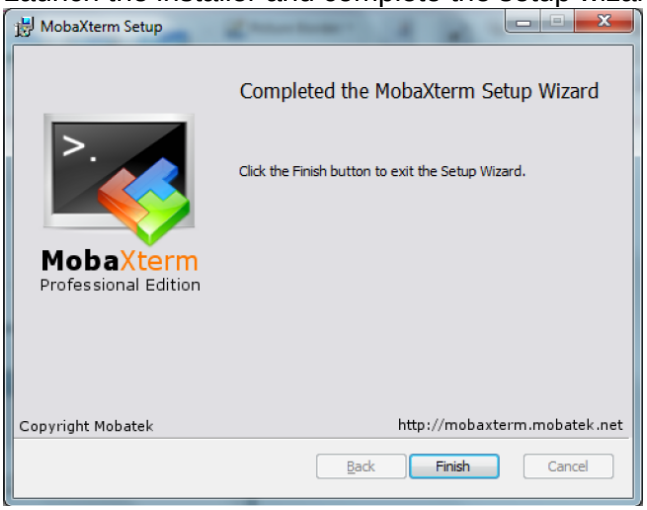
Verify the content of the *timestwo_rtlabsfcn_package* folder to make sure there is no intellectual property. You can now send it to the party who will integrate the S-Function into a RT-LAB model.

ANNEX 1: MOBAXTERM INSTALLATION AND CONFIGURATION

MobaXterm will be used in order to transfer the source files of the S-Function to the target and execute commands. It is a free program. We will be using the SSH and SFTP client functionalities.

INSTALLATION OF MOBAXTERM

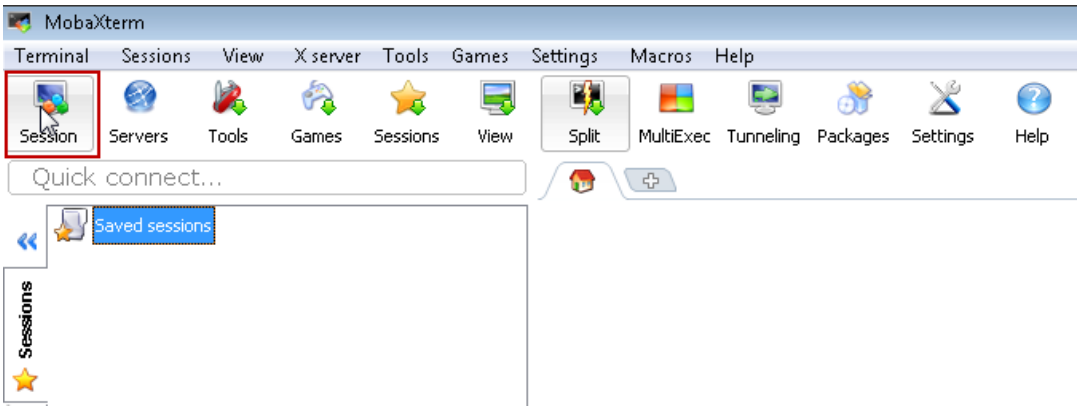
1. Download MobaXterm: http://mobaxterm.mobatek.net/download.html.
2. Launch the installer and complete the setup wizard.



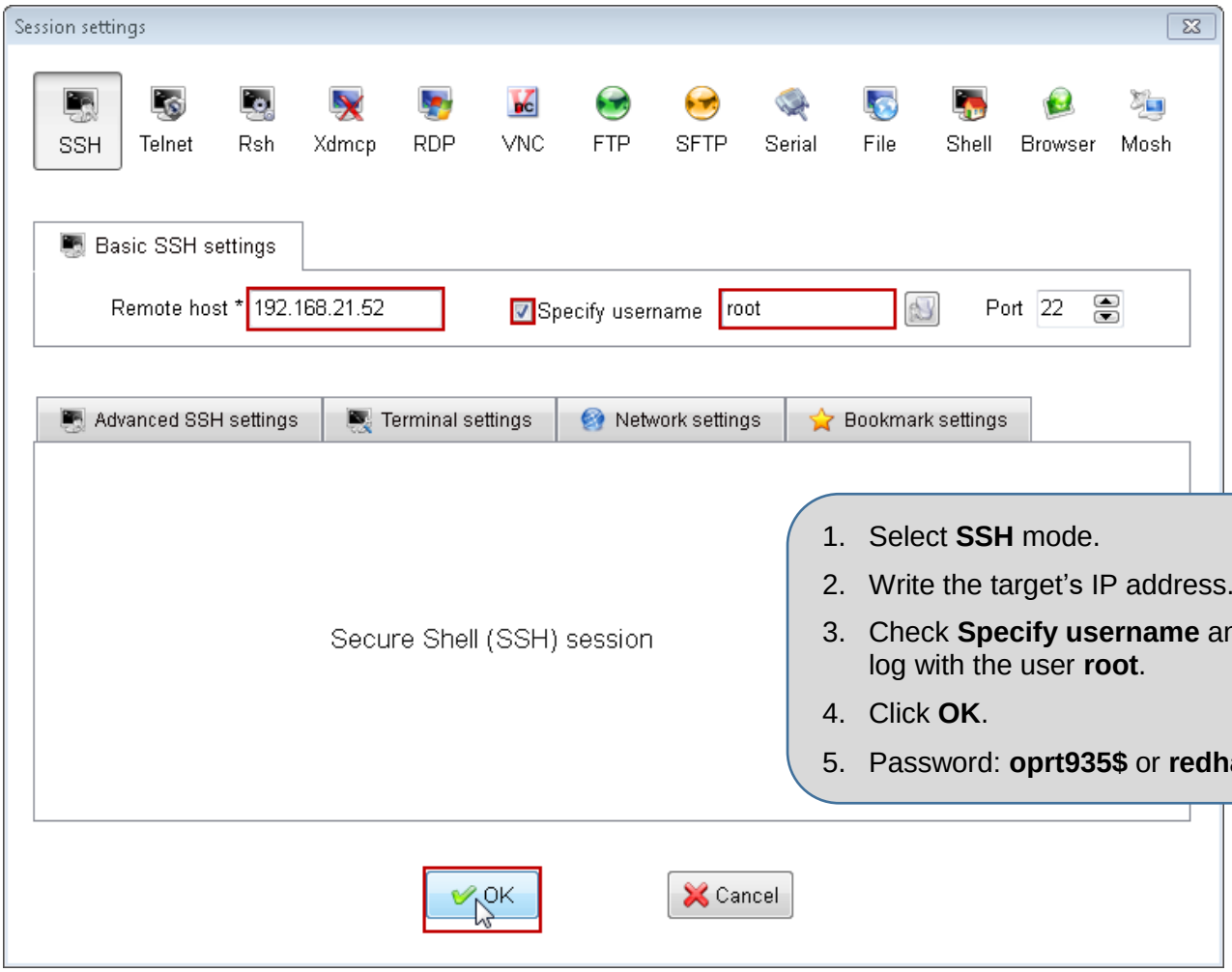
CONFIGURATION OF MOBAXTERM

3. Open MobaXterm

4. Click **Session** to create a new session.



5. Configure the new session:



1. Select **SSH** mode.
2. Write the target's IP address.
3. Check **Specify username** and log with the user **root**.
4. Click **OK**.
5. Password: **oprt935$** or **redhat**

6. When the session configuration is completed, a session starts automatically. For future use, open a session by double clicking on the session list on the left side of MobaXterm.

Technical Services
www.opal-rt.com/support

CONTACT

Opal-RT Corporate Headquarters

1751 Richardson, Suite 2525
Montréal, Québec, Canada
H3K 1G6
Tel.: 514-935-2323
Toll free: 1-877-935-2323

Note:

While every effort has been made to ensure accuracy in this publication, no responsibility can be accepted for errors or omissions. Data may change, as well as legislation, and you are strongly advised to obtain copies of the most recently issued regulations, standards, and guidelines.

This publication is not intended to form the basis of a contract.