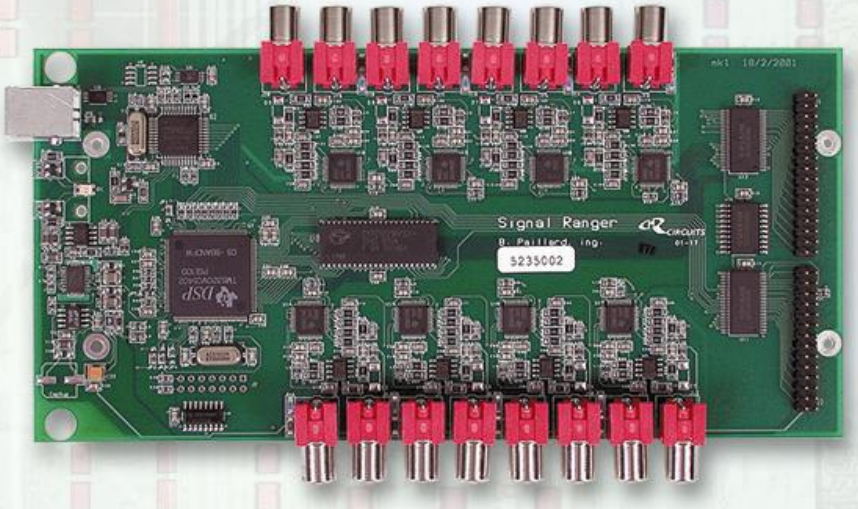


Simulink to Signal Ranger

Converter

By Soft dB inc.

March 2011

Version 2.2 March 2011

Table of content

Table of figures

1. Simulink To Signal Ranger Converter Description

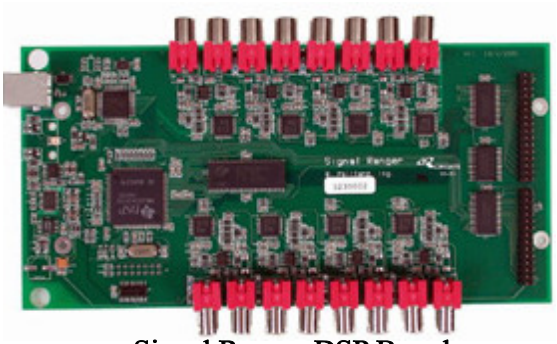
1.1 Overview

The software SimulinkToSR Converter now allows you to easily implement graphical MatLab/Simulink design onto a Signal Ranger DSP board (SR1) or the more recent Signal Ranger MK2 board (SR2). Once you have completed your controller under Simulink, you can convert and implement your design within a few simple clicks.

1.1 Features

SIMULINK®
Dynamic System Simulation for MATLAB®

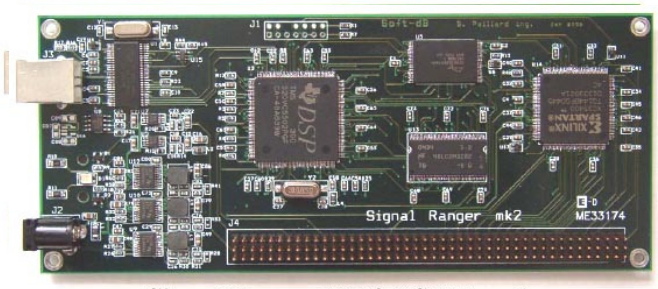
- Code conversion from MatLab/Simulink to Signal Ranger DSP board.
- Simple user interface: convert, load & execute on DSP, read and write on DSP.
- Modifications of user defined variables through the interface in real time.



Signal Ranger DSP Board

1.2 Applications

- Motor controller
- General multi-channels controller
- Modern control
- Signal processing on audio signal
- …



Signal Ranger MK2 DSP Board

1.3 Requirements

- Signal Ranger DSP board (SR1) or Signal Ranger MK2 board (SR2)[1] by **SoftdB**[2]
- MatWorks R13 (MatLab 6.5) or MatWorks R14 (MatLab 7.0), including Simulink and Real-time Workshop[3]
- Code Composer Studio software[4] by Texax Instruments.

---

[1] A SR2_Analog expansion board is also required in order to access a maximum of 16 analog IOs.

[2] Refer to www.softdb.com for more details about the Signal Ranger DSP board and other products.

[3] SimulinkToSR Converter has been tested under MATLAB version 6.5(R13), 6.5.1(R13SP1), 7.0.1(R14SP1) and 7.0.4(R14SP2).

[4] SimulinkToSR Converter has been developed under *Code Composer Studio version 2, version 3.3 and version 4* (see Annexe A on How to get Code Composer Studio).

- Windows 2000 or newer

1.4 Software Installation

Download the software installer from www.softdb.com/dsp-products-simulink.php

1- Run the self-extracting executable file SimulinkToSRX_Installer_vX.exe

2- Follow the installer instructions

You should now have access to the program from a shortcut in **Start Menu→ Programs→ SimulinkToSR→ SimulinkToSRX**. The current software documentation can also be accessed from the same location.

The executable file is installed by default in \Program Files\SimulinkToSRX\. Moreover, the working directory of the application is \ProgramData\SimulinkToSRX\. In this directory, you can find the documentation file, this example directory as well as the IncludeSources directory. The IncludeSources directory will eventually contain copies of the files from a third party protected by copyrights[5].

1.5 Demo version limitations

The demo version of the software is provided as a 3-month trial period. Once this trial completed, we invite you to contact Softdb for a program upgrade.

Most of the main Simulink blocks are already supported by our application. Consult the Annexe B for the list of the blocks tested so far with the SimulinkToSR converter. At this moment, blocks with continuous time are not supported. Other non-enumerated blocks are not guaranteed to work.

For the DSP on the Signal Ranger board, the code size is limited 32 Kbytes for SR1 board and 64 Kbytes for SR2 board. At the time of the compilation, our converter will present a linking error if the code size is too large. In such a case, you would have to reduce the size of your controller, but it's possible to achieve a relatively complex controller before you reach the limit.

Up to 100 MIPS (Millions of instructions per second) are allowed on SR1 board and up to 300 MIPS on SR2 board. Following the complexity of the Simulink diagram and the sampling frequency used, you can meet computation time limits of the DSP. To help you in this matter, the SimulinkToSR interface displays the timing requirements in the upper left. Obviously, your application becomes critical when "Time usage" approach 100% and will crash as it goes over.

1.6 Credits

This project has been initiated at the Université de Sherbrooke for a project of speciality by the following students: Mathieu Hamel and François Gagnon. They demonstrated the functionality of the concept and they established the base of the current software. Soft dB owes them special thanks.

[5] See the 2.2.1.1 section on Required files for more details

2. Implementing Simulink design on Signal Ranger DSP

The program developed by SoftdB takes advantage of the Real-Time Workshop utility available in Simulink. The implementation process from Simulink to the DSP then goes in two distinct parts:

2.1 Creation of Simulink model

2.2 Conversion and implementation with SimulinkToSR interface

2.1 Creation of Simulink model

To facilitate the setup of Simulink model, we provide you a simple example that is already configure to build a model with Real-Time Workshop. We suggest you to begin your design from the example located in the "example" directory. It includes 4 files that you can copy and rename as needed (except for grt_rtw_info_hook.m that have to keep the same name):

example.mdl : the pre-configured Simulink file

example_init.m : the file that initiates variables

example_sim.m : the simulation file (optional)

grt_rtw_info_hook.m: Real-Time Workshop needs that file to generate the model in relation to the information specific to the DSP. This file has to be present in the same directory as the Simulink file.

With Simulink, there are 4 main steps to convert Simulink design into DSP code.

2.1.1 Configure your simulink file

2.1.2 Design your system under Simulink

2.1.3 Predefine Simulink variables in MatLab workspace

2.1.4 Simulate your system

2.1.5 Build model with Real-Time Workshop

2.1.1 Configure your simulink file

If you are starting from the example.mdl (or a copy), go directly to the next subsection.

Note that unless mentioned otherwise, the default simulation parameters of simulink are used.

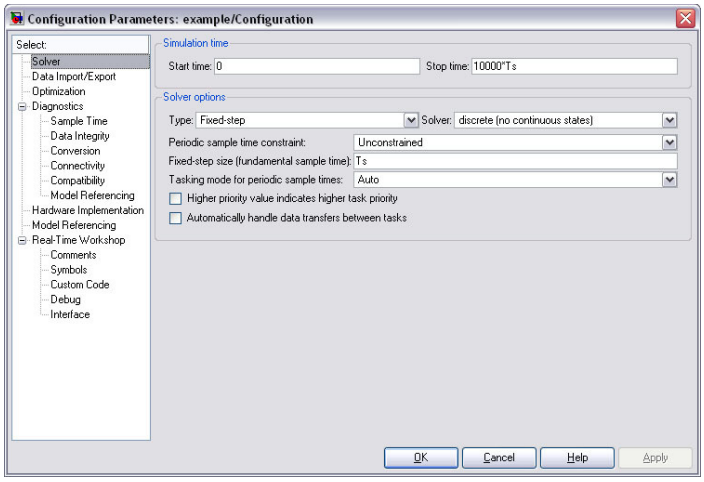


Figure 1: Setting Solver options in Simulink

From your Simulink file, go to **Simulation→ Configuration Parameters→ Solver**.

Simulation time parameters can be useful if you want to simulate your system inside Simulink, but are not relevant for now.

In the Solver options section, select the type “Fixed-step” and the solver “discrete(no continuous states)”. Also set the *Fixed step size* to the sample time you want to use. We suggest always using the variable “**Ts**” which have to be previously defined in MatLab workspace (see 2.1.3 section).

With the SR1 board, the sampling frequency is restrained to Fs = 21700/N, where N = {1, 2, 3, …, 32}. With the SR2 board, the are many possible values from 286.102 Hz to 24414.062 Hz (the list is available in Annexe C).

Knowing the Fs value, “Fixed step size” can be fixed to Ts = 1/Fs.

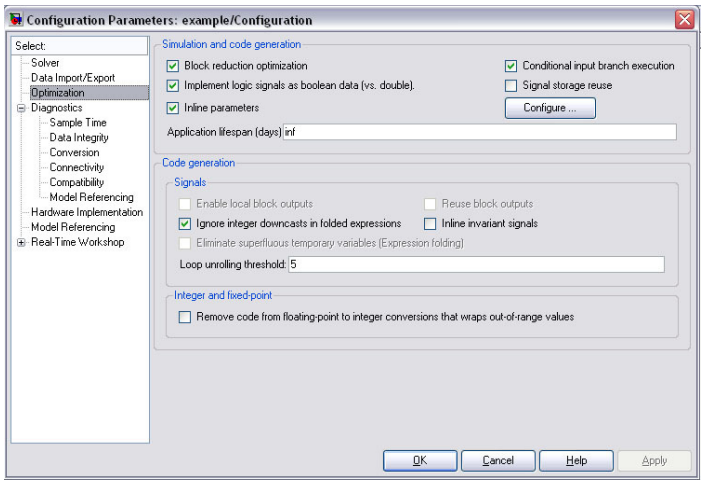


Figure 2: Enabling Inline Parameters

Go to **Simulation→ Configuration Parameters→ Optimization**.

In the Simulation and code generation section, enable *Inline parameters*.

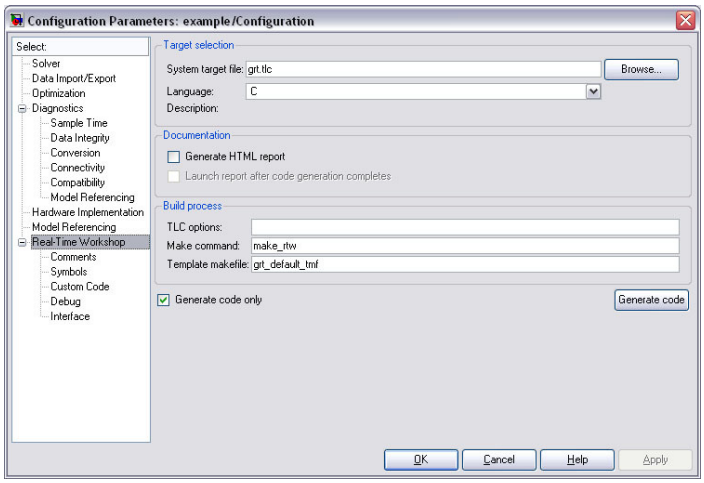


Figure 3: Setting Real-time Workshop options

Go to **Simulation→ Configuration Parameters→ Real-Time Workshop**.

Enable *Generate code only*.

2.1.2 Design your system under Simulink

Use Simulink help as needed if you are not familiar with the software.

Consult the list in the Annexe B, to know which blocks are so far supported by the current version of the SimulinkToSR converter.

Note that input and output amplitudes in Simulink directly represent voltage amplitudes found on the board IOs. Also, you should know that the analogue dynamic ranges are normally of +/- 10 volts for the inputs and +/- 2 volts for the outputs. All values over those ranges will saturate to their maximal value on the board.

Let's now explore different possibilities through different block types used in the example.mdl file.

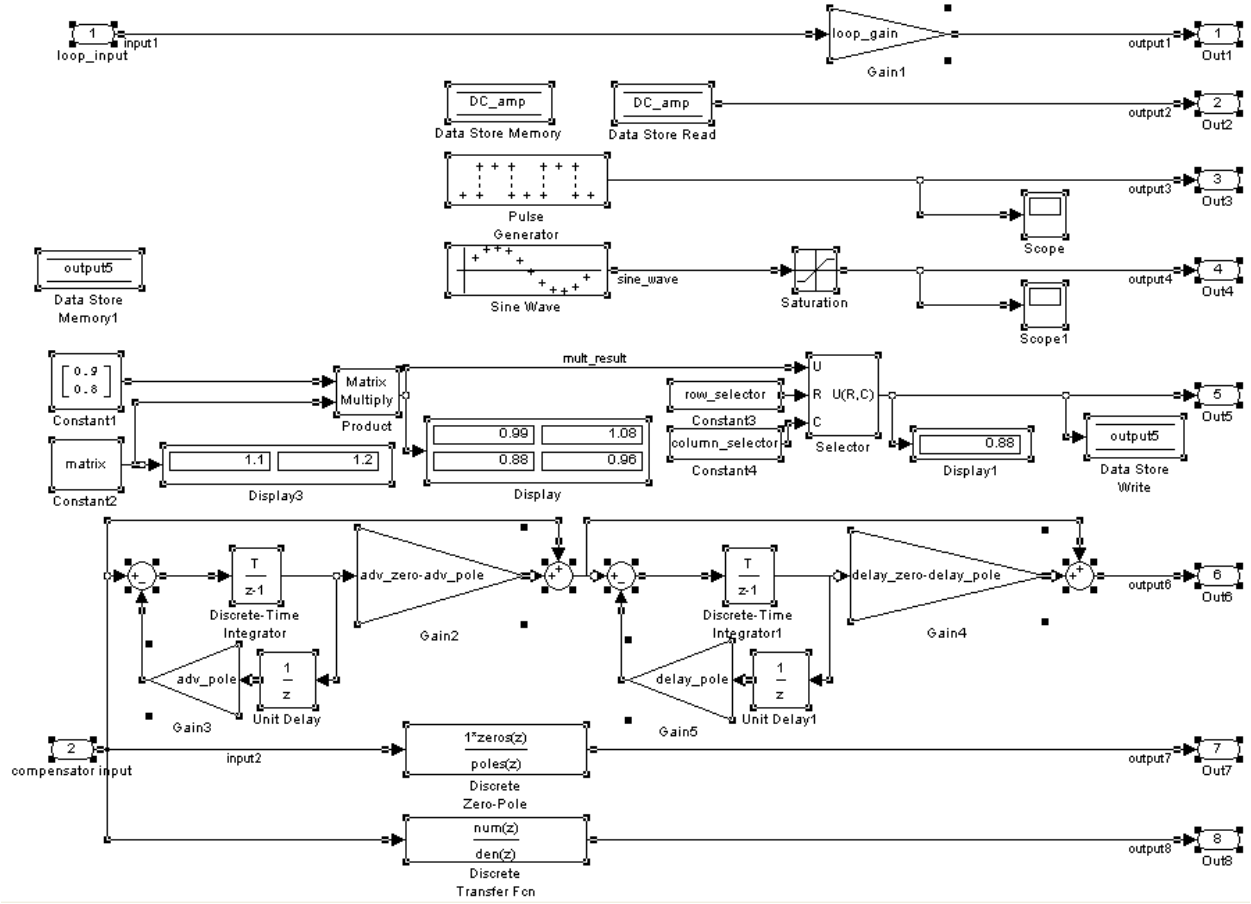


Figure 4: Example Simulink file, Example.mdl

If the number of Inport and Outport in the Simulink design exceeds the physical number of analog IOs on the board, the processor will still process the operations related to a virtual IO. Also, port numbers of Inport and Outport blocks are directly related, in ascending order, to the available IOs of the board.

The Simulink example showed in figure 4 illustrates a variety of blocks and configurations. The first port is in loopback, so unity loop_gain should result in an analogue output of same voltage as the input. DC, pulse and sinusoid sources are send to outputs 2, 3 and 4 respectively. Matrix operations are used to send a value to output 5. Finally, three different ways to compute a phase compensator are used for outputs 6, 7 and 8.

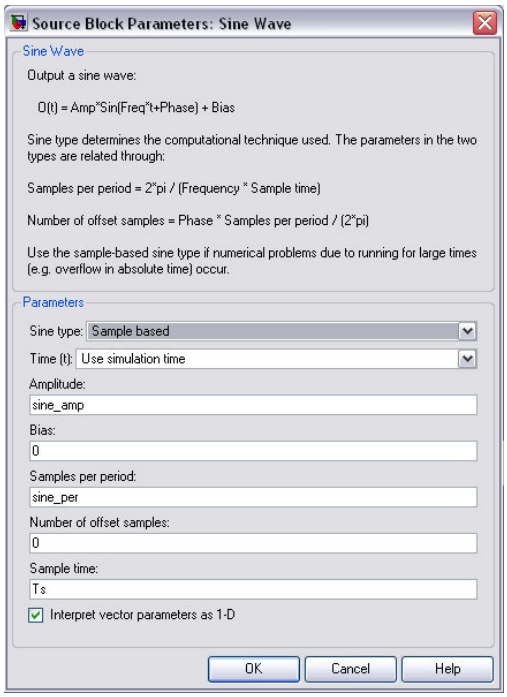


Figure 5: Setting Sine Wave parameters

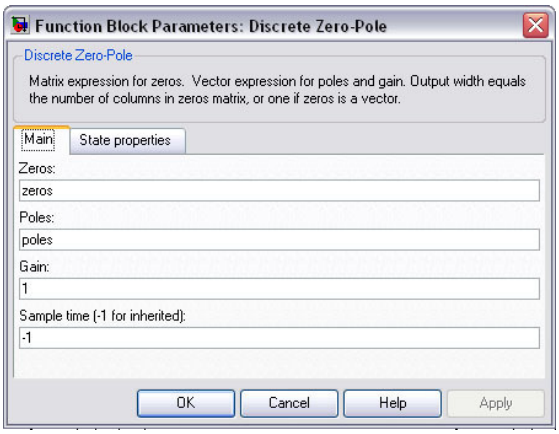


Figure 6: Setting Discrete Zero-Pole parameters

When the *Sample Time* parameter is present in block parameters, you should assign its value to the same as the *Fixed step size* set in previous subsection (figure 5). When it is mentioned beside the parameter, you can alternatively set its value to "-1" for inherited (figure 6).

As seen in the examples, several parameters are set symbolically through variables. We will later see that it is possible to modify most of block parameters dynamically once the program is implemented[6]. To do so, use variable names instead of numbers, vectors or matrixes. Of course, those variables have to be previously defined in MatLab workspace (see next subsection).

2.1.3 Predefine Simulink variables in MatLab workspace

All variables assigned to Simulink parameters must be declared before Simulink can use them. To ease the process, you should use a M-file[7] as an initiation file.

For now, you can look the example_init.m as an example.

[6] If for some reason, some parameters cannot be modified in real-time, Real-Time Workshop usually mentions it when it builds the model. Moreover, those parameters won't be accessible through the SimulinkToSR interface.

[7] For MatLab neophytes, a M-file is a file in which you can write MatLab code and then call it in MatLab. You can create one from MatLab→new→M-file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% example_init.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File that initiates the user defined variables used in the example.mdl
% Simulink file. This init file is intended for both SR1 and SR2 board.
% By default, the parameter of frequency Fs is set for SR1 while the
% parameter of Fs of SR2 is in comment.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% By SoftdB, june 6th 2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SELECT HERE THE SAMPLING PERIOD Ts that should be used for the "fixed step size"
% parameter set in SIMULINK-->Simulation-->Configuration Parameters-->Solver
% and that should also be used for every block sample time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% enable Fs of SR1 or Fs of SR2 below

% For SR1 board:
N=26;                   % Select one of the restrained value of N={1,2,3,...,32}
Fs=N/21700;             % Resulting Simulink sampling period (seconde)

% For SR2 board:
%Fs=832.298;             % Select a frequency allowed by SR2 board
                         % You may set the aimed frequency and SimulinkToSR converter
                         % will provide you the closest frequency allowed.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resulting Simulink sampling period (seconde). One of the Fs above must be uncommented.
Ts=1/Fs;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%     Definitions of variables used in Simulink file     %%%%%%%%%%%%%%%%%%%
loop_gain = 1;          % default loopback gain on the first AIC
f_sine = 1.0;           % frequency (hertz) for the sinusoid
sine_amp = 1;           % Default sinusoid amplitude (volts)
sine_per = round(1/(Ts*f_sine));% Default sinusoid period (samples), frequency(Hz)=1/(Ts*sin_per)
pulse_amp   = 1;        % Default pulse amplitude (volts)
pulse_per   = sine_per; % Default pulse period (samples), frequency(Hz)=1/(Ts*pulse_per)
pulse_width = round(pulse_per/2);% Default pulse width (samples), DutyCycle=pulse_width/pulse_per
matrix = [1.1 1.2];     % Matrix 1 by 2
```

Figure 7: Example initiation M-File, example_init.m

Note that the previously used sampling period "Ts" is defined in this initialization file. Its value is set through the constant "N" which suggested values should be respected.

Once you are done writing your M-file, execute it in MatLab[8]. If you forget to declare some variables the simulation or the Real-time Workshop builder will further detect it.

2.1.4 Export Variables

With the Signal Ranger DSP board, it is possible to communicate with DSP while the code is running. It is then possible to read and write dynamically parameters called "ExportedGlobal" in your Simulink model. This feature gives the designer great flexibility.

There are three ways to access user variables:

---

[8] The easiest way to execute a M-file is by pressing "F5". You can alternatively set the "Current Directory" parameter in MatLab to your M-file directory and then enter your M-file name (without extension) in the Command Window.

2.1.4.1 Predefined variables

You can access dynamically the symbolic Simulink parameters set with variables in Matlab workspace (see previous subsection). However, you have to specify which variables have to be exported. Indeed, all variables you want to access in real-time have to be configured "ExportedGlobal" in Simulink.

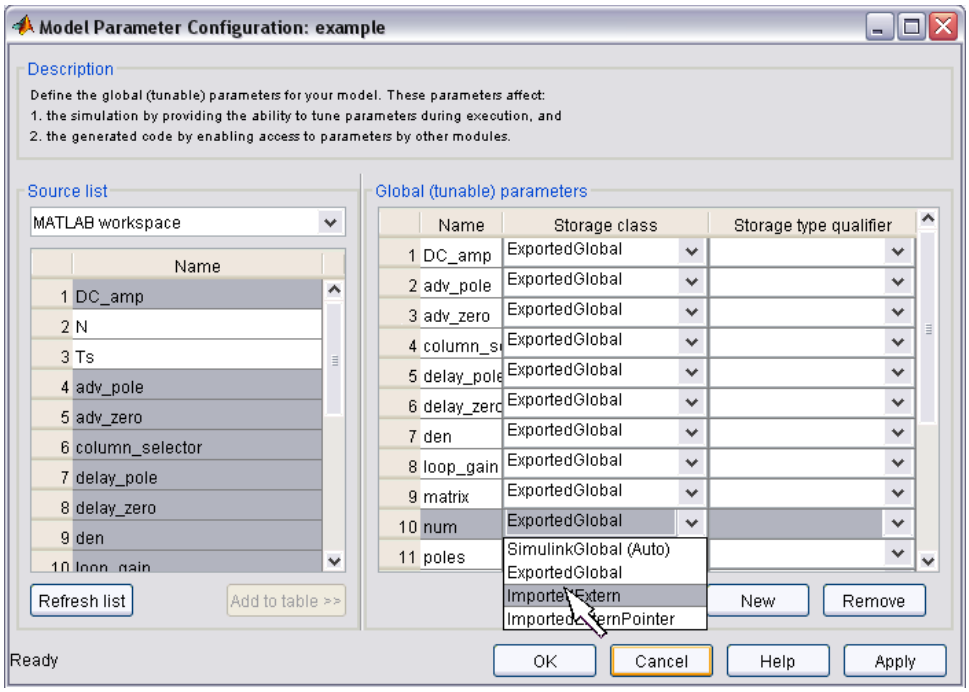


Figure 8: Set variables "ExportedGlobal"

To do so:

1- From your Simulink file, go to **Simulation→ Configuration Parameters→ Optimization→ Configure**. All variables found in MatLab workspace are enumerated in the Source list (figure 8).

2- Select the variables you wish to access through the SimulinkToSR interface and press *add to table>>*. The added variables then appear in *Global (tunable) parameters*.

3- Set the *Storage class* of all the added variables to “ExportedGlobal” and press OK to confirm.

This works only if the variable is actually used somewhere as a parameter of your system. Moreover, some variables will be ignored if Simulink does not allow a particular parameter to be dynamic.

2.1.4.2 Data Store blocks

The Simulink blocks Data Store Memory, Data Store Read and Data Store Write allow you to respectively declare, read and write a data variable. Simply set the *RTW Storage Class* parameter of the Data Store Memory block to "ExportedGlobal" to give a dynamic access through the interface (figure 9).

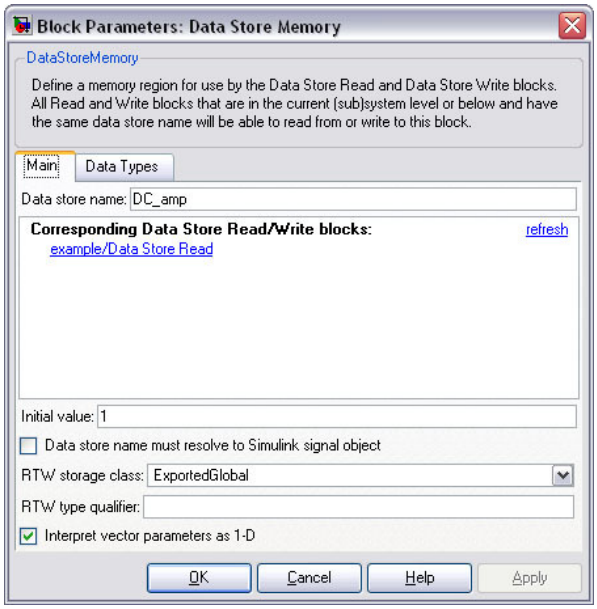


Figure 9: Exporting DataStoreMemory for DC_amp

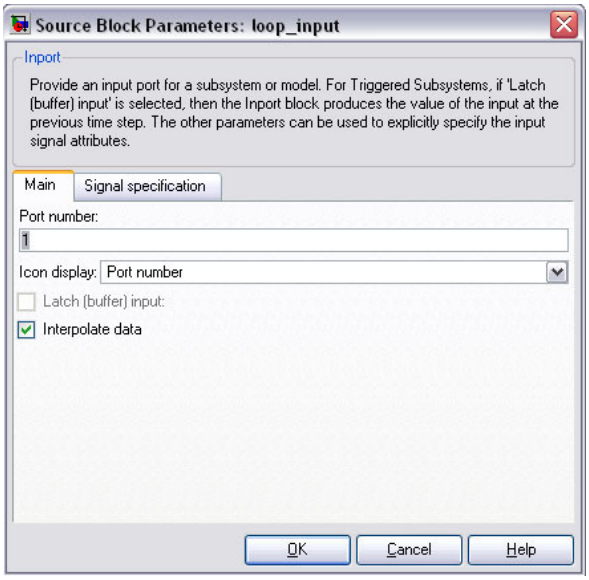


Figure 10: Exporting Wire Probe for input1

2.1.4.3 Wire Probes

The use of a probe on a wire allows you to read (only) its value through the interface. To do so, right-click on the wire, select "Signal Properties..." and set the *RTW Storage Class* parameter to "ExportedGlobal" (figure 10). Avoid setting ExportedGlobal wires going to Outport block especially if you are using MATLAB 7.0(R14) because it may cause the Real-Time Workshop to ignore assignment to Outport (this issue was not fixed yet in R14SP2).

2.1.5 Simulate your system

Simulations are obviously not necessary, but it's usually much simpler to debug your system under the MatWorks environment than once the program implemented on the DSP. The present document is not really intended to show you how to simulated, but let's quickly see what are the options.

One way to do your simulation is to use the tools directly in Simulink. As in the example, you can place Scope and Display blocks on the signals of interest. Afterwards, you can start the simulation by going to **Simulation→ Start** (or start icon).

One more flexible way is to insert the model in a M-file that will simulate it. You can then simulate the system inputs and plot the outputs among other things.

To incorporate you Simulink mode into a M-File, you need to use the "sim" MatLab function. Its main input parameters are the Simulink model filename, the time span and an input table that describes the inputs in function of sample time. The function then generates the outputs. Read example_sim.m for an example of implementation. By executing this simulation file, you can see if the previous model example works as expected.

2.1.6 Build model with Real-Time Workshop

Finally, build the model with Real-Time Workshop from you Simulink file.

Go to Tools→ Real-Time Workshop→ Build Model (likewise Ctrl+B or build icon).

If no error is detected, the building process appears in the MatLab command window. Warnings are also displayed there. In the example, warnings tell you that the Discrete Zero-Pole parameters will be set constant (correspond to the block comments in Annexe B).

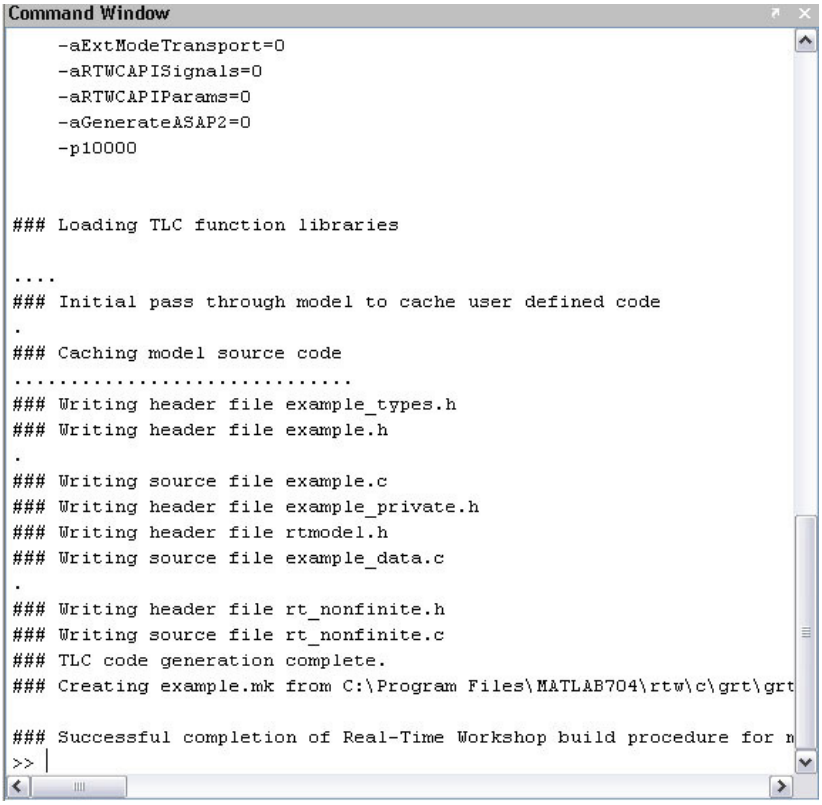


Figure 11: Real-Time Workshop build process of the example

A successful build of your Simulink system creates a new directory of name *model*_grt_rtw (*model* being your Simulink file). The files inside that directory will then be used by the SimulinkToSR converter to implement the system onto a Signal Ranger DSP board.

2.2 Conversion and implementation with SimulinkToSR interface

The SimulinkToSR converter has the task to convert the Real-Time model created in Simulink and to implement the system on the DSP.

Open the SimulinkToSR converter and follow the instructions.

Here is description of the interface:

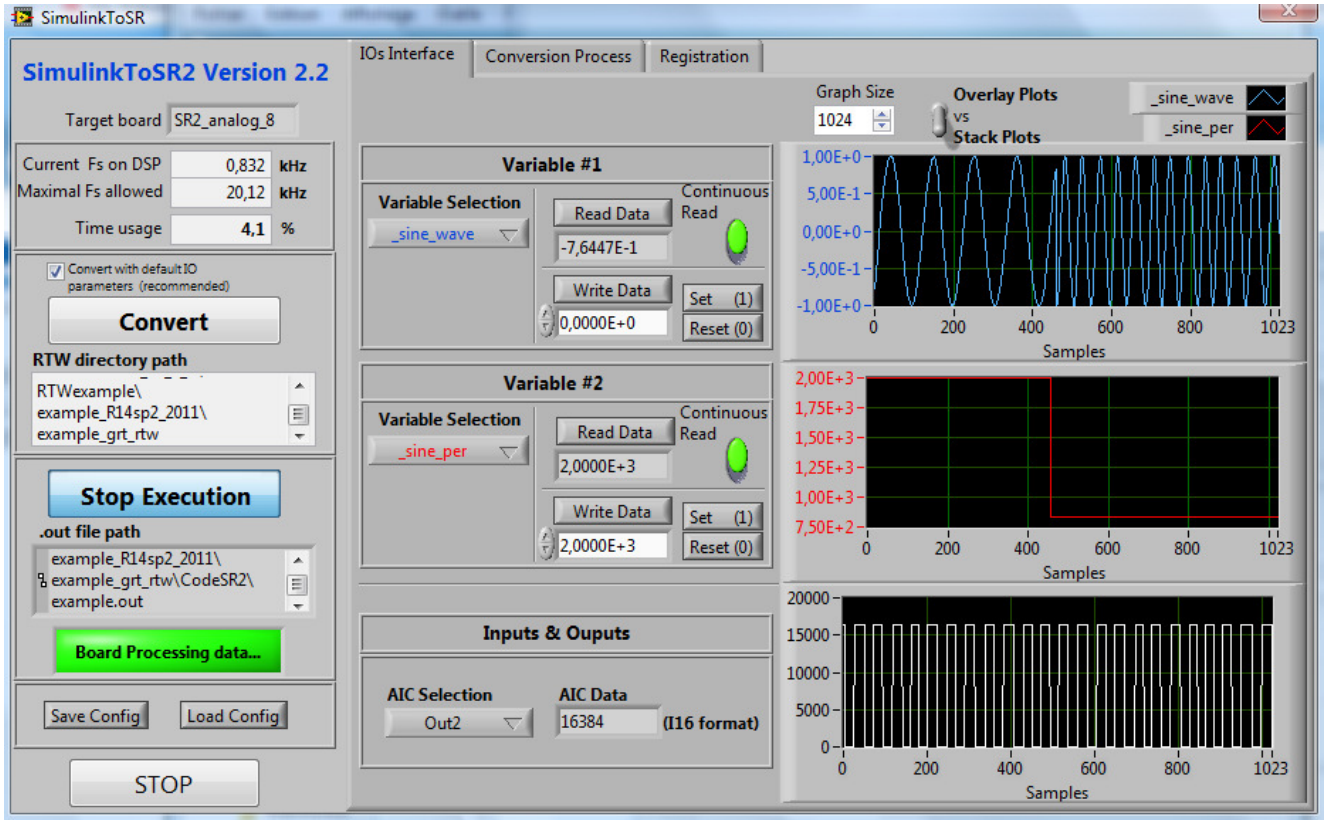


Figure 12: SimulinkToSR interface

2.2.1 Convert button

By pressing on the *Convert* push button, you start the conversion process of the code generated from Real-Time Workshop (under Simulink) to a .out file that can be implemented on the Signal Ranger DSP board.

The conversion process can be resumed through the following elements:

1- AIC parameters

2- Real-Time Workshop file

3- Sampling frequency check

4- Required files

5- Conversion Process textbox

2.2.1.1 AIC parameters

A first window appears if *Convert with default IO parameters* checkbox is unchecked. This window allows you to set the AIC parameters of the board, which are the IOs of the Signal Ranger DSP board.

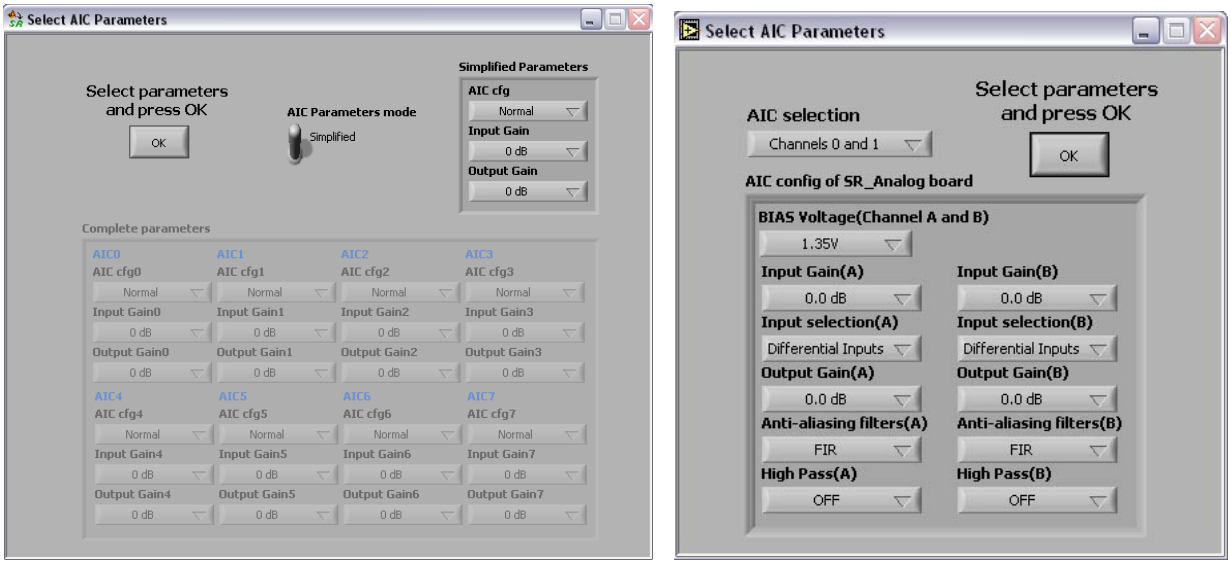


Figure 13: AIC parameters setting panel for SR1 and for SR2

You can set a gain to the AIC input and output with *Input gain* and *Output gain*. This is particularly useful to enhance the ratio signal on noise (SNR) when the input signal is low. However, care should be taken when applying gains. There is normally an equivalence between the analogue voltages on the AICs and the input and output amplitudes in the simulink model. This correspondence is no longer valid when gains are used. Let's recall that dynamic ranges are normally (gains of 0 dB) of +/- 10 volts for the inputs and +/- 2 volts for the outputs.

SR1

Switching the AIC parameters mode in «Simplified» or «Complete» position respectively to set AICs individually or globally. You can set AICs "Normal", "Analog loopback" or "Digital loopback" by using *AIC cfg* parameter. Unless specific aims are intended, the "normal" option is recommended.

SR2

Each AIC of SR2 board contains 2 channels. The parameters will be set accordingly. Also, SR2 board exist in different configuration that doesn't contains the same number of AICs. Therefore, only the existing AICs will be allow in the AIC selection menu. Once again default options are recommended unless specific aims are intended.

2.2.1.2 Real-Time Workshop file

Once all required files are found, SimulinkToSR converter asks you to identify the Real-Time Workshop directory built from Simulink model (RTW directory path). This directory can normally be found in the Simulink model directory and the name should be in the format *model*_grt_rtw.

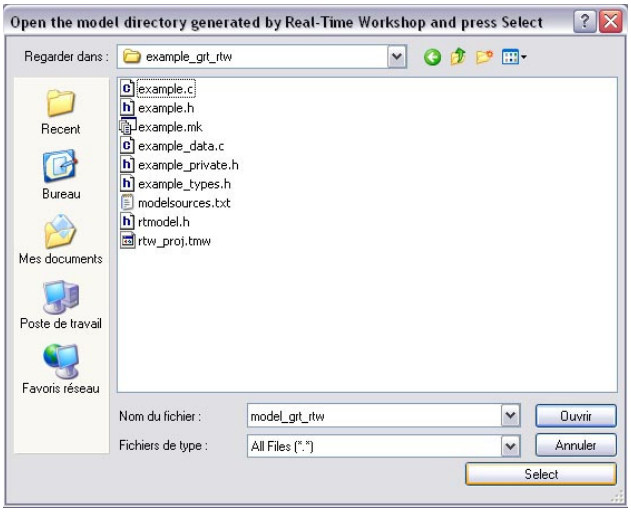


Figure 14: Selection of Real-Time Workshop directory in the example

2.2.1.3 Sampling frequency check

If the Simulink "fixed step size" does not match one of the available sampling frequencies on the DSP, a window appears. A warning is displayed and we suggest you the closest frequency that would match the frequency set in Simulink. To disable this window, it is recommended to set the "fixed step size" in Simulink to an allowed value (see section 2.1.1).

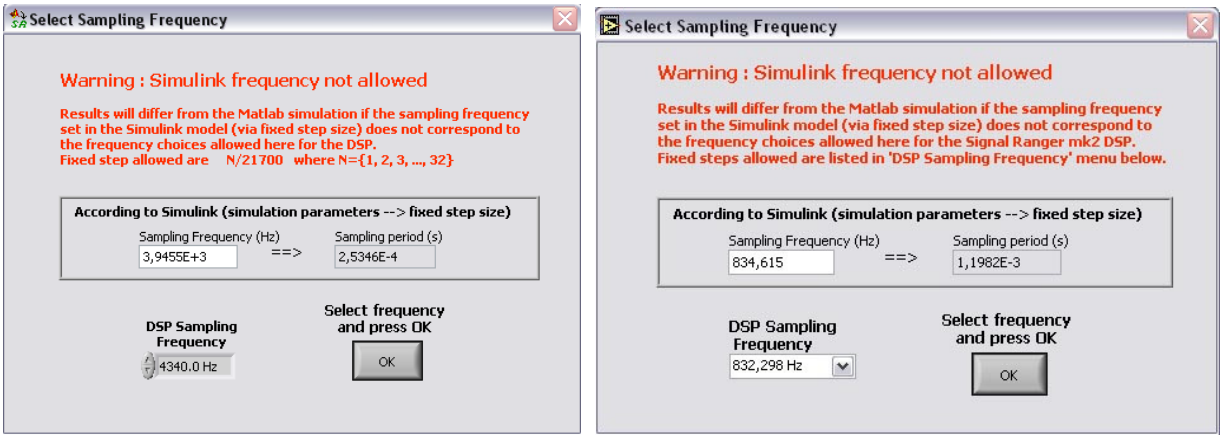


Figure 15: Frequency check window for an invalid Simulink sampling frequency for SR1 and for SR2

2.2.1.4 Required files

The first time the SimulinkToSR program runs on your computer, you may be asked to provide some files.

In order to respect copyrights of third software companies, you need to provide some files to the current program. The SimulinkToSR program normally copies Texas Instruments files in 'IncludeSources\TI_files' directory and MatWorks files in 'IncludeSources\MatWorks_files' directory. The program will fail if some files are missing. The complete file list can be found in Annexe D.

MatWorks files should not cause any problem because they belong in majority to Real-Time Workshop, which you used previously.

Texas instruments files are part of the Code Composer Studio that should be installed on your computer. If you do not possess the TI files, go to Annexe A on How to get Code Composer Studio.

2.2.1.5 Conversion Process textbox

During all the conversion process, the current operation is displayed in the textbox of the *Conversion Process* tab (figure 16). Errors and warnings are also displayed in this textbox.

As the process completes, a led appears at the bottom of the window. A green led indicates that the conversion is successful and a red led indicates that the conversion failed.

If no mistake occurred, you are invited to load and execute the program on the DSP.

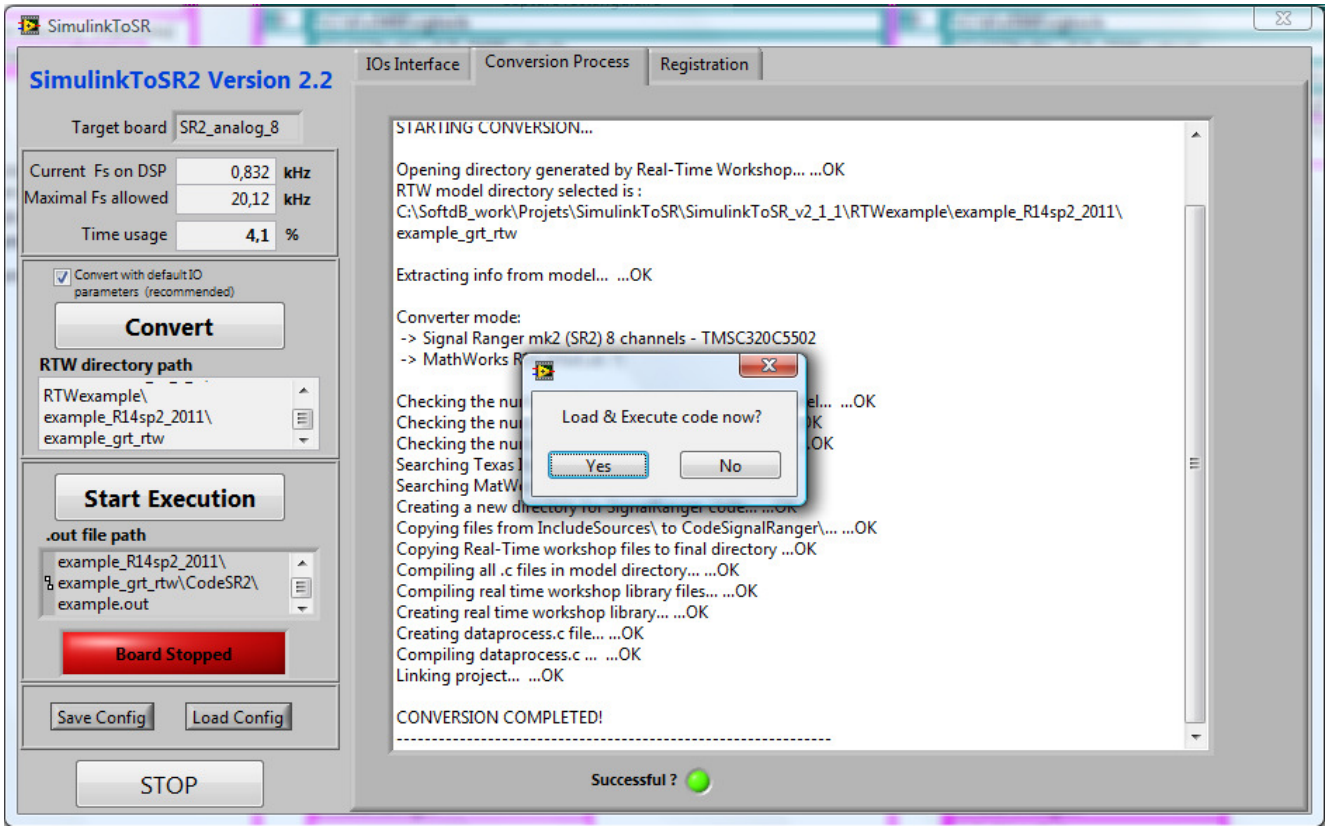


Figure 16: Conversion Process

Because code size is limited by the DSP, the linker may display an allocation error if the code size is larger than the limit.

2.2.2 Start/Stop Execution button

By switching the *Start/Stop Execution* button, you can start and stop the execution of DSP program. Therefore, .out file is reloaded every time you press *Start Execution* and processing is interrupted when you press *Stop Execution*. Reading and writing obviously end when the execution stops.

Under the execution button, a rectangular led lets you know if the DSP board is properly running or not according to a green or a red light.

2.2.3 Processing Time Usage

A higher sampling frequency on the DSP means that it can process fewer operations in one cycle. However, the relation between design complexity and processing time is not obvious. Therefore, the interface displays the rate of time usage. In fact, *Time Usage* indicates its higher value since the last time the code has been executed. This way, you can better judge if the timing requirement is critical and if you can increase the sampling frequency.

2.2.4 Save and Load configuration

A normal use of the program requires converting your Simulink model at least once before loading it on the board. You can then set a few parameters to display the variables of interest. You might find useful the *Save Config* and *Load Config* buttons in order to start with the configuration of previous session. Doing so, you can start the execution without converting your model again.

2.2.5 Variable #1 and Variable #2 interfaces

The *Variable #1* and *Variable #2* of the *IOs Interface* tab allow you to display and modify their value on the DSP.

By using *Variable Selection* menu, you can select a variable you wish to display or modified. An underscore precedes the name of your MatLab variable names.

The program supports Simulink arrays and matrixes. Therefore, an *Index* can be applied on the variable DSP address if you need to access elements of a vector or a matrix variable. The *Index* and *Length* parameters will be visible only when the variable is really an array. Particular attention should be taken while using index on matrix variable. Once on the DSP, matrixes are converted in vectors by appending columns beginning by the leftmost one.

By pressing *Read* push button, you read a single sample of the selected variable. You can alternatively read in continuous by switching on the *Continuous Read* button. Read values are displayed on both the *Read Data* and on the plot on the right.

By pressing *Write* push button, you write the *Write Data* value to the selected variable. You can alternatively use the *Set* and *Reset* button to directly write 1 or 0 respectively.

2.2.6 Inputs & Outputs interface

The *Inputs & Outputs* interface of the *IOs Interface* tab allows you to display the numeric value of the AICs (the inputs and outputs of the board).

By using *AIC Selection* menu, you can select the channel you wish to display. Let's remember that port numbers of Inport and Outport Simulink blocks are directly related, in ascending order, to the available IOs of the board.

Read values are displayed on both the *AIC Data* and on the plot on the right[6]. Since AIC are represented in 16 bits signed format, the values range from –32768 to 32767. The numeric range normally matches the analogue ranges, respectively of +/- 10 volts for AIC inputs and +/- 2 volts for AIC output[9].

2.2.7 Signal waveforms

Three graphics are showed on the *IOs Interface* tab; Variable #1, Variable #2 and one IO.

Variables 1 and 2 can be displayed in overlay mode or stack mode by switching up or down the button above the plots.

The number of displayed samples can be adjusted through the *Graph Size* parameter. Its value can vary from 1 to 1024 samples.

Note that data are read in an asynchronous sampling period of 10 ms or higher. Therefore, the waveforms on the interface will not necessarily represent the exact waveforms the DSP generates. Also, trying to link samples with time will not be precise.

2.2.8 *Stop* button

Press the *Stop* push button to end the program. If the code was running on the DSP, it will not be interrupted.

[9] Analogue ranges will differ if you use gains on the AIC parameter panel during the conversion process.

Annexe A: How to get Code Composer Studio

In order to respect Texas Instruments copyrights, you need to provide Code Composer Studio files to our program. The SimulinkToSR program normally copies automatically the required files in the 'IncludeSources\TI_files' directory.

Let's first tell you how to get the required TI files:

1- Purchase a DSP Starter Kit (DSK). This will give you a version of Code Composer Studio that will work indefinitely with your DSK. It will also allow you to provide the necessary files for our Simulink/Signal Ranger converter. Use the link below to purchase a TMS320VC5416 DSK, if you would like to take this option.

   TMS320VC5416 DSP Starter Kit (for use with our Signal Ranger DSP board (SR1)) :

   http://www.spectrumdigital.com/product_info.php?cPath=24_64&products_id=100&osCsid=5a157cecae4a7338adf02278de155432

   TMS320VC5510 DSP Starter Kit (for use with our Signal Ranger MK2 board (SR2)):

   http://www.spectrumdigital.com/product_info.php?cPath=24_64&products_id=108&osCsid=5a157cecae4a7338adf02278de155432

2- Download your 120-day trial evaluation tools for C5000 DSPs. It includes a version of Code Composer Studio. It will also allow you to provide the necessary files for our Simulink/Signal Ranger converter.

   Here is the hyperlink to the Code Composer Studio™:
   http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html

3- Purchase the full version of Code Composer Studio.

Annexe B: List of supported Simulink blocks

Here is the list of the blocks tested so far with the SimulinkToSR converter. Non-enumerated blocks are not guaranteed to work.

Unless otherwise specified all block parameters can be modified in real-time once implemented. To achieve that, set block parameters with “ExportedGlobal” variables, which have to be predefined in MatLab workspace (see sections 2.1.3 and 2.1.5 for more details)

Group	**Block name**	**Functionality** 1 = Perfect 2 = OK with limitations 3 = To come in further version 4 = Not supported by SimToSR 5 = Not supported by RTW	**Comments**
Simulink/Continuous	*All Blocks of group*	5	**Not Supported** by RTW becauce Continuous
Simulink/Discontinuities	Backlash	2	"InitialOutput" parameter cannot be controlled in real-time (RTW limitation)
Simulink/Discontinuities	Coulomb & Viscous Friction	1	
Simulink/Discontinuities	Dead Zone	1	
Simulink/Discontinuities	Hit Crossing	1	
Simulink/Discontinuities	Quantizer	1	
Simulink/Discontinuities	Rate Limiter	4	This block generates a new sample time of 0 (continuous) which is impossible to achieve.
Simulink/Discontinuities	Relay	1	
Simulink/Discontinuities	Saturation	1	
Simulink/Discrete	Discrete transfer function	1	
Simulink/Dicrete	Discrete poles-zeros	2	Parameters cannot be controlled in real-time (RTW limitation)
Simulink/Dicrete	Discrete filter	1	
Simulink/Dicrete	Discrete state-space	1	
Simulink/Dicrete	Discrete-Time Integrator	1	
Simulink/Dicrete	Memory	1	
Simulink/Dicrete	Unit Delay	1	
Simulink/Math Operations	Abs	1	
Simulink/Math Operations	Algebraic constraint	5	Real-Time Workshop Error: Real-Time Workshop does not support models with algebraic loops.
Simulink/Math Operations	Assignment	1	Rows and columns elements have to be external if it is desired to modified them in real-time.
Simulink/Math Operations	Bitwise Logical Operator	1	
Simulink/Math Operations	Combinatorial Logic	2	Do not modify "truth table" parameter dynamically.
Simulink/Math Operations	Complex to Magnitude-Angle	1	
Simulink/Math Operations	Complex to Real-Imag	1	
Simulink/Math Operations	Dot Product	1	

Simulink/Math Operations	Gain	1	
Simulink/Math Operations	Logical Operator	1	
Simulink/Math Operations	Magnitude-Angle to Complex	1	
Simulink/Math Operations	Math Function	3	Are OK : exp, 10^u, magnitude^2, square, sqrt, pow, conj, reciprocal, mod, transpose, hermitian,hypot **Not supported yet**: **log, log10,rem**
Simulink/Math Operations	Matrix Concatenation	1	
Simulink/Math Operations	Matrix Gain	1	
Simulink/Math Operations	MinMax	1	
Simulink/Math Operations	Polynomial	1	
Simulink/Math Operations	Product	1	
Simulink/Math Operations	Real-Imag to Complex	1	
Simulink/Math Operations	Relational Operator	1	
Simulink/Math Operations	Reshape	1	
Simulink/Math Operations	Rounding Function	1	
Simulink/Math Operations	Sign	1	
Simulink/Math Operations	Slider Gain	1	
Simulink/Math Operations	Sum	1	
Simulink/Math Operations	Trigonometric Function	2	Are OK : sin, cos, tan, asin, acos, atan, cosh, sinh, tanh, atan2 **asinh, acosh, atanh** are **not supported** are not supported by ANSI-C (do not use)
Simulink/Ports & Subsystems	Enable	1	
Simulink/Ports & Subsystems	Enabled Subsystem	1	
Simulink/Ports & Subsystems	Enabled and Triggered Subsystem	1	
Simulink/Ports & Subsystems	If	1	
Simulink/Ports & Subsystems	If Action Subsystem	1	
Simulink/Ports & Subsystems	In1	1	
Simulink/Ports & Subsystems	Out1	1	
Simulink/Ports & Subsystems	Subsystem	1	
Simulink/Ports & Subsystems	Trigger	1	
Simulink/Ports & Subsystems	Triggered Subsystem	1	
Simulink/Signal Attributes	Data type conversion	1	

Simulink/Signal Routing	Data Store Memory	1	Set properties to "ExportedGlobal" to access data dynamically in SimulinkToSR interface
Simulink/Signal Routing	Data Store Read	1	
Simulink/Signal Routing	Data Store Write	1	
Simulink/Signal Routing	Demux	1	
Simulink/Signal Routing	Multiport Switch	1	
Simulink/Signal Routing	Mux	1	
Simulink/Signal Routing	Selector	1	The “indices” parameters have to be external in order to be modified in real-time.
Simulink/Signal Routing	Switch	1	
Simulink/Sinks	Display	1	Simulation tool only… ignored by RTW
Simulink/Sinks	Out1	1	*See also Simulink/Ports & Subsystems*
Simulink/Sinks	Scope	1	Simulation tool only… ignored by RTW
Simulink/Sinks	Terminator	1	
Simulink/Source	Band limited white Noise	4	**Not supported**
Simulink/Source	Chirp Signal	4	**Not supported**
Simulink/Source	Clock	5	**Not supported**
Simulink/Source	Constant	1	
Simulink/Source	Digital clock	4	**Not supported**
Simulink/Source	From workspace	4	**Not supported**
Simulink/Source	From file	4	**Not supported**
Simulink/Source	Ground	1	
Simulink/Source	In1	1	*See Simulink/Ports&Subsystems/In1*
Simulink/Source	Pulse generator	1	Select "Sample based" in the "Pulse type" parameter of the block
Simulink/Source	Ramp	4	**Not supported**
Simulink/Source	Random Number	4	**Not supported**
Simulink/Source	Repeating Sequence	4	**Not supported**
Simulink/Source	Signal generator	4	**Not supported**
Simulink/Source	Signal builder	4	**Not supported**
Simulink/Source	Sine wave	1	Select "Sample based" in the "Sine type" parameter of the block
Simulink/Source	Step	4	**Not supported**
Simulink/Source	Uniform Random Number	4	**Not supported**

Annexe C: List of frequency supported by the SR2 board

Values are in hertz.

286.102	409.747	508.626	631.398	807.076	1101.386	1575.101	2547.554	5859.375
290.644	410.320	510.399	634.131	813.802	1105.542	1583.615	2569.901	5978.954
295.331	413.798	512.183	636.889	820.641	1109.730	1592.221	2615.792	6103.516
300.173	415.559	513.980	638.276	827.595	1122.486	1600.922	2639.358	6233.378
305.176	416.149	516.700	642.475	832.298	1126.803	1609.718	2663.352	6368.886
310.020	417.334	519.448	643.887	834.669	1131.153	1627.604	2712.674	6510.417
310.348	418.527	522.226	649.598	837.054	1135.538	1655.191	2763.856	6658.381
315.020	420.932	523.158	651.042	841.864	1144.409	1664.595	2790.179	6813.227
315.699	422.754	524.094	653.948	849.185	1148.897	1674.107	2817.007	6975.446
320.184	424.592	525.034	659.840	851.653	1157.979	1683.728	2872.243	7145.579
321.238	425.209	527.872	662.825	854.136	1162.574	1703.307	2929.687	7324.219
325.521	425.827	530.740	664.328	856.634	1171.875	1713.268	2959.280	7512.019
326.974	427.068	532.670	665.838	859.146	1181.326	1723.346	2989.477	7709.704
331.038	428.317	533.641	673.491	861.673	1186.108	1733.543	3051.758	7918.074
332.164	429.573	536.573	675.043	866.771	1190.930	1743.862	3083.882	8138.021
332.919	433.386	538.545	678.168	871.931	1195.791	1775.568	3116.689	8370.536
336.746	434.028	542.535	681.323	879.786	1200.692	1786.395	3150.202	8616.728
337.522	435.965	543.541	682.911	887.784	1205.633	1808.449	3184.443	8877.841
339.084	436.615	549.660	686.109	890.483	1210.615	1819.682	3219.437	9155.273
342.654	441.883	550.693	690.964	893.197	1220.703	1831.055	3255.208	9450.605
343.055	443.892	551.730	692.598	901.442	1230.961	1842.571	3329.190	9765.625
345.482	445.241	552.771	697.545	904.225	1241.393	1878.005	3367.457	10102.371
348.772	446.599	554.865	700.882	909.841	1246.676	1890.121	3406.613	10463.170
352.126	450.721	558.036	704.252	915.527	1252.003	1902.394	3446.691	10850.694
354.684	451.416	561.243	707.654	918.397	1262.796	1914.828	3487.723	11268.029
355.114	452.112	563.401	709.367	921.285	1268.263	1927.426	3572.790	11718.750
357.715	454.215	565.577	714.558	930.060	1273.777	1953.125	3616.898	12207.031
359.030	454.920	566.671	718.061	939.002	1284.951	1979.519	3662.109	12737.772
360.799	457.764	567.769	719.825	945.060	1302.083	1992.985	3756.010	13316.761
361.690	459.199	571.089	721.598	951.197	1307.896	2020.474	3804.789	13950.893
363.485	459.920	572.205	723.380	957.414	1319.679	2034.505	3854.852	14648.437
366.211	460.643	574.449	726.970	960.553	1325.650	2048.733	3906.250	15419.408
367.129	465.030	577.848	732.422	963.713	1331.676	2077.793	3959.037	16276.042
368.514	467.255	578.990	734.258	973.318	1350.086	2092.634	4069.010	17233.456
369.444	469.501	581.287	739.820	976.562	1356.337	2122.962	4185.268	18310.547
373.685	472.530	585.937	747.369	979.829	1362.645	2154.182	4245.924	19531.250
375.601	475.599	590.663	751.202	986.427	1381.928	2170.139	4308.364	20926.339
380.479	476.372	591.856	757.025	989.759	1395.089	2202.773	4438.920	22536.058
381.470	478.707	593.054	760.958	993.114	1401.764	2219.460	4507.212	24414.062
381.967	479.491	595.465	762.939	996.492	1408.504	2253.606	4577.637	
382.966	480.277	597.895	770.970	1010.237	1415.308	2271.076	4650.298	
387.525	481.856	600.346	775.050	1017.253	1429.116	2288.818	4725.302	
388.553	484.246	602.816	777.105	1020.797	1436.121	2325.149	4802.766	
389.586	486.659	605.307	779.172	1024.366	1443.196	2343.750	4882.812	
390.625	488.281	609.083	781.250	1027.961	1464.844	2362.651	4965.572	
393.775	489.914	610.352	783.339	1038.896	1479.640	2381.860	5051.185	
394.837	493.213	614.190	787.550	1046.317	1494.739	2401.383	5139.803	
395.369	494.880	615.481	789.673	1050.067	1502.404	2421.229	5231.585	
398.055	496.557	619.384	791.807	1061.481	1525.879	2441.406	5326.705	
398.597	498.246	620.697	793.953	1065.341	1541.941	2461.922	5425.347	
400.231	500.801	623.338	796.111	1073.146	1550.099	2482.786	5527.712	
402.430	502.519	626.002	800.461	1077.091	1558.344	2504.006	5634.014	
406.901	505.119	630.040	804.859	1085.069	1566.678	2525.593	5744.485	

Annexe D: List of files required for the converter

The SimulinkToSR converter will be looking for the following files related to your Matlab version.

MatWorks files needed with Matlab 6.5 (R13) Default path: C:\MATLAB6p5\	MatWorks files needed with Matlab 7.0 (R13) Default path: C:\Program Files\MATLAB704\
rtw\c\libsrc\rt_mxclassid.h rtw\c\libsrc\rt_logging.h rtw\c\libsrc\rtlibsrc.h rtw\c\libsrc\rt_matmultrr_dbl.c rtw\c\libsrc\rt_matmultandincrr_dbl.c rtw\c\libsrc\rt_atan2.c rtw\c\libsrc\rt_hypot.c rtw\c\libsrc\rt_zcfcn.c extern\include\tmwtypes.h simulink\include\simstruc_types.h	rtw\c\libsrc\rt_mxclassid.h rtw\c\libsrc\rt_logging.h rtw\c\libsrc\rtlibsrc.h rtw\c\libsrc\rt_matmultrr_dbl.c rtw\c\libsrc\rt_matmultandincrr_dbl.c rtw\c\libsrc\rt_atan2.c rtw\c\libsrc\rt_hypot.c rtw\c\libsrc\rt_zcfcn.c extern\include\tmwtypes.h simulink\include\simstruc_types.h simulink\include\rtw_matlogging.h simulink\include\rtw_extmode.h simulink\include\rtw_continuous.h simulink\include\rtw_solver.h simulink\include\sysran_types.h simulink\include\simstruc.h

The SimulinkToSR converter will be looking for the following files related Code Composer Studio from TI. Files will differ whether you are using SR1 or SR2 board.

Texas Instruments Files needed for SR1 board Default path… CCS2: C:\ti\c5400\cgtools\ CCS3: C:\CCStudio_v3.3\c5400\cgtools\ CCS4: C:\Program Files\Texas Instruments\ccsv4\tools\compiler\c5400\	Texas Instruments Files needed for SR2 board Default path… CCS2: C:\ti\c5500\cgtools\ CCS3: C:\CCStudio_v3.3\c5500\cgtools\ CCS4: C:\Program Files\Texas Instruments\ccsv4\tools\compiler\c5500\
cgtools\lib\rts.lib cgtools\bin\acp500.exe cgtools\bin\ar500.exe cgtools\bin\asm500.exe cgtools\bin\cg500.exe cgtools\bin\cl500.exe cgtools\bin\lnk500.exe cgtools\include\float.h cgtools\include\limits.h cgtools\include\linkage.h cgtools\include\math.h cgtools\include\string.h	cgtools\lib\rts55.lib cgtools\bin\acp55.exe cgtools\bin\ar55.exe cgtools\bin\asm55.exe cgtools\bin\cg55.exe cgtools\bin\cl55.exe cgtools\bin\lnk55.exe cgtools\bin\masm55.exe cgtools\include\float.h cgtools\include\limits.h cgtools\include\linkage.h cgtools\include\math.h cgtools\include\mathf.h cgtools\include\mathl.h cgtools\include\string.h cgtools\include\unaccess.h cgtools\include\access.h