



### **Quick User Guide**

As its name indicates, the SimulinkToSR program links Matlab/Simulink files to the <u>Signal Ranger DSP board</u>. Doing so, DSP programming is now possible for almost everyone.

First, the user needs Matlab/Real-Time Workshop to transform the Simulink model into C language.

Next, the generated files are sent through the SimulinkToSR Converter. It, then, generates a .out file that can be directly implemented on the Signal Ranger board.

The required process time ranges from a few seconds to one minute depending on the system complexity.

After these steps have been completed, the SimulinkToSR interface allows you to communicate with the DSP board. It is then possible to dynamically modify the parameters of your Simulink system and see its effect right away.

#### SimulinkToSR Converter Demonstration Video

Download the demo video that explains how to use the SimulinkToSR Converter software and shows the main functionalities.

# Simulink Demo Video

#### **Implementation Procedure**

The following text describes the quick steps to implement a Simulink design on Signal Ranger DSP. However, the reading of the user manual is recommended for better comprehension.

- 1. Configure your Simulink file
- 2. Design your system under Simulink
- 3. Predefine Simulink variables in MatLab workspace
- 4. Export variables
- 5. Build model with Real-Time Workshop
- 6. Convert and implement with SimulinkToSR interface

# 1. Configure Your Simulink File

To facilitate the setup of Simulink model, we provide you a simple example that is already configured 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 the sampling period (Ts) and other user 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.

# 2. Design Your System Under Simulink

Use the Simulink help as needed if you are unfamiliar with the software.

Consult the list in Annexe B found in user manuel to know which blocks are 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, take note that the analogue dynamic ranges are normally of  $\pm 10$  volts for the inputs and  $\pm 2$  volts for the outputs. All values over these ranges will saturate to their maximal value on the board.



When the *Sample Time* parameter is present in block parameters, assign its value to the same as the *Fixed step size* (Ts). When it is mentioned beside the parameter, its value can alternatively be set to "-1" for inherited.

# 3. Predefine Simulink Variables in MatLab Workspace

Simulink parameters can be assigned a variable set in Matlab Workspace. To ease the process, use a *M-file* as an initiation file before running the Simulink file. For now, you can look at *example\_init.m* as an example.

<u> </u>						-
~ I.	<	A PERSONA DELOS DE	THUSE NEVELISE	and three verdy		5
26 - 32	den = [-1, 0, -1, 0];	h Defeult Foles of	Phage Advance	and mase Delay		
24	< rnase compensator p	Anameters, compact to	Discos Adversa	or and denumerator		
	zecos = [1.0 1.0];	* Derault Zeros of	rnase Advance	and Phase Delay		
2	pores = [1.0 1.0];	a perault roles of	rhase Advance	and mase berey		
11	- rnase compensator p	arameters, compact to	the with poles	S MALE AND		
- 0	delay_pole = 1.0;	A Default Pole of	Phase Delay			
19	celay_zero = 1.0;	* percult Zero of	rnase Delay			
18 -	adv_pole = 1.0;	* Default Pole of	Phase Advance			
7 -	adv_zero = 1.0;	4 Default Zero of 1	Phase Advance			
16	? Phase compensator p	arameters, extensive	Cora			
25 -	row_selector=2;	A Default row selection	ction			
4 -	column_selector=1;	% Default column s	election			
13 -	matrix = [1.1 1.2];	8 Matrix 1 by 2	the second s			
22 -	pulse_width = kound(p	ulse_per/2);% Default	pulse width (:	samples), DutyCycle-	-pulse_width/p	ti l
21 -	pulse_per = sine_pe	r; A Default pu	ise period (sou	ples), frequency(Hs	z)=1/(Ts*pulse	3
20 - 0	pulse_amp = 1;	> Default pulse amplitude	plitude (volts)	Nata desta - companya de martes		
9 -	sine_per = round(1/(T	s*f_sine));% Defoult :	sinusoid period	i (samples), frequer	ncy(Hz)=1/(Ts*	31
8 -	sine_amp = 1;	4 Default sinusoid	amplitude (vo)	lts]		
7 -	f_sine = 1;	+ frequency (hertz	) for the sinus	016		
6 -	loop_gain = 1;	% default loopback	gain on the fi	irst AIC		
15	\$\$\$\$\$\$\$\$\$\$	efinitions of variable	es used in Sim	link file 💠 👯	*****	*
4	*******	******	***********	**************	*****	\$
3 -	Ts=N/21700;	A Resulting Simuli:	nk sampling per	ciod (seconde)		
2 -	N=5;	A Select one of th	e restrained va	alue of N={1,2,3,	,32}	
1	and that should als	o be used for every b	lock sample tit	ie		
0	6 parameter set in SI	WULINK>Simulation:	Simulation Par	consters>Solver		
9	& SELECT HERE THE SAM	PLING PERIOD that show	uld be used for	the "fixed step so	ize"	
8	*****	****	QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ	*****	******	ħ,
7	By Soft@B, june 15t	h 2004				
6	*****	****	****	***************	*****	ş
5	<pre>% Simulink file</pre>					
4	File that initiates	the user defined var.	iables used in	the example.mdl		
3	***************	*************	*****	***************	*****	÷
2	<pre>% example_init.m</pre>					
4		****	****	****		2
41 1					and a state of the second s	

Note that the previously used sampling period "Ts" is also defined in this initiation file.

### 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 dynamically read and write parameters called "ExportedGlobal" in your Simulink model.

There are three ways to access user variables:

#### Declare "ExportedGlobal" Predefined Variables

You can dynamically access the Simulink parameters set to variables in Matlab workspace. However, the variables to be exported must be specified. Indeed, all variables to be accessed in real-time have to be configured "ExportedGlobal" in Simulink.

To do so:

- From your Simulink file, go to **Simulation > Simulation Parameters > Advanced > Configure**. All variables found in MatLab workspace are enumerated in the *Source list*.
- 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*.
- Set the Storage class of all added variables to "ExportedGlobal" and press OK to confirm.

Description Define the global (tunable) parameter 1. the simulation by providing the abit 2. the generated code by enabling ac	s for your mo ity to tune pa cess to param	del. These parameters rameters during exect eters by other module	affect: rtion, and R.		
Source list		Olobal (tunable)	parameters		
MATLAB workspace 💉		Name	Storage class	Storage type qualifier	8
Alama		1 DC_amp	ExportedGlobal 🗸	~	1
1 DC ome	~	2 adv_pole	ExportedGlobal 🗸	~	1
1 DG_amp	-	3 adv_zero	ExportedGlobal 🗸	~	1
2 70		4 column_s	ExportedOlobal 🗸 🗸	v	1
3 15	10	5 delay_pola	ExportedGlobal 🗸	×	1
4 adv_pole	2.1	6 delay_zero	ExportedGlobal 🗸	~	1
6 soluton colorier		7 den	ExportedOlobal 🗸 🗸	v	1
7 delev eele	-	8 loop_gain	ExportedOlobal 🗸	~	1
7 delay_pore		9 matrix	ExportedGlobal 🗸	~	1
6 delay_zero	-	10 num	ExportedGlobal 🗸	×	1
10 loon rain		11 poles	SimulinkGlobal (Auto) ExportedGlobal	*	
Refresh list Add to	table >>		Imported Extern Imported emPointer	New Remove	

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.

#### **Use 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 dynamic access through the interface.

#### **Use 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".

#### 5. Build Model with Real-Time Workshop

To 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.

```
Command Window
                                                                            R X
### Generating code into build directory: .\example_grt_rtw
Varning: Reducing run-time parameter 'A' of block 'example/Discrete Zero-Pole'
Warning: Reducing run-time parameter 'C' of block 'example/Discrete Zero-Pole'
Warning: Reducing run-time parameter 'D' of block 'example/Discrete Zero-Pole'
### Invoking Target Language Compiler on example.rtw
tic -r .\example grt rtw\example.rtw 'C:\MATLAB6p5\rtw\c\grt\grt.tic' -0.\exam
### Loading TMW TLC function libraries
### Initial pass through model to cache user defined code
### Caching model source code
### Writing header file example.h
### Uriting header file example_types.h
### Writing source file example.c
### Writing header file example_private.h
### Writing header file rtmodel.h
### Writing source file example_data.c
### TLC code generation complete.
### example.mk which is generated from C:\MATLAB6p5\rtw\c\grt\grt_lcc.tmf is up
### Successful completion of Real-Time Workshop build procedure for model: exam
>>
<
                                                                            5
```

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

# 6. Convert and Implement with SimulinkToSR Interface

The SimulinkToSR Converter is responsible for converting the Real-Time model created in Simulink and for implementing the system on the DSP.

			3Os Interface Convers	ton Process			
Current: Ps on DSP	4,34	kHz			Graph See 256 👲	Overlay Plots	joutputt
Maximal Fs allowed 5,606 MHz		kHz	Vari	ables #1	1,00240-		الكراك الأخلاك
Tine usage	11,4	*	Variable Selection	Real Data Continue	9,00E-1-		
Convert with default 10			Crie and T	5,000E-1	7,005-1-		
Conv	ert			Write Data Set (	1) S.ODE-1-		
RTW file path	1000		i	Strateset 0	0 25	50 75 000 125 11 Samples	0 175 005 271 0
C/(Projet_stage)		Vari	iables #2	7,508-1-			
evanole)	- interest	1	Variable Selection	Read Data Contract	5,006-1 - 2,522-1 - 0,922+0 -		
Stop Exe	cutio	n	Copier User Variables Only (recommended)	Provided .	2,506-0-		
.out file path		aw	Neoroscie and and a	Write Data Set (	7,500-1-		
CritProjet_stage)	0	-		glo,00000+0 Reset (	0 25	50 75 100 125 15 Sangles	0 175 200 225 3
Smulniktexanole	i ja	8	77-		1,502+4 -		
Board Processing datas			Inputs & Ouputs		L,00E+4- 5,00E+3-	AAA	ΛΛΛ
Save Carring	Land Cor	io	AIC Selection	AIC Data 5314 (116 form	0,00E+0- 5,00E+3- -1,00E+4-	VVVV	IVVV
STO	P	1	1		-1,50€++	60 75 100 125 19	0 175 200 225 3

Open the SimulinkToSR Converter, press the **Convert** button, and follow the instructions. Within a few seconds, your Simulink Design should be ready to execute.

You can now interact with the Signal Ranger DSP board through the SimulinkToSR interface.