

# How to Use PSIM-Embed Wrapper Link Altair PSIM Tutorial

ALTAIR © Altair Engineering, Inc. All Rights Reserved. / altair.com / Nasdaq: ALTR / Contact Us

# Introduction

PSIM-Embed Wrapper generates a DLL block for controller implementations inside PSIM and exports it to Embed so that the user can generate target code for any DSP that Embed supports. The generated wrapper blocks using the PSIM subcircuit can be used inside Embed's schematic environment. The DLL file will be automatically registered inside Embed under its main menu named "Wrapper Blocks". Each DLL file has one or multiple blocks under a sub-menu named "My Blocks for \*\*\*", where \*\*\* is the name of the PSIM subcircuit.

## Step-by-step wrapper implementation

This tutorial will guide users on how to generate a wrapper DLL and use the generated wrapper blocks inside Embed.

#### Step 1: Open a PSIM schematic example having a wrapper subcircuit.



Step 2: Right mouse click on the "Wrapper Subcircuit". Select "Attributes" from the pop-up menu.



Next, a dialog will be displayed as shown below to "Generate Code".

Subcircuit : S1	×
Change Subcircuit File	Subcircuit Variables Formula Fixed-Point Color
Subcircuit	Help
Variable Description	Variable Name Variable Value
	<b>1</b>
	-
	•
Add	Modify Remove
Add	Modify Remove
Add Set as Default Varia	Modify Remove Des Reload Default Variables with generated code for simulation

Step 3: Click on the "Generate Code" button and select "Generate Wrapper Code" (as shown below).

This will generate a wrapper folder at this example path destination. The wrapper folder's name will be the subcircuit's file name + (C code), as shown below.

$local Disk (C:) \rightarrow$	PSIM 2022.3.0.48 >	examples >	Code Generation >	Wrapper >	BuckConverter	>
	F 211VI 2022-2-0-40 /	examples /	Code Generation /	wiappei /	DUCKCONVEILER	1

Name	Date modified	Туре	Size
subBuck (C code)	3/28/2023 8:20 PM	File folder	
BuckConverter.psimsch	3/17/2023 9:43 PM	PSIMSCH File	61 KB
🛃 BuckConverter_Embed.psimsch	3/23/2023 7:16 PM	PSIMSCH File	60 KB
陀 BuckConverter_Hardware.vsm	3/17/2023 9:43 PM	Vissim32 Document	12 KB
🛃 subBuck.psimsch	3/17/2023 9:43 PM	PSIMSCH File	16 KB

For example, suppose that one wants to generate wrapper code for subcircuit S1 and the subcircuit's file name is Sub2Count.psimsch. Then, the code is generated under the same folder as the current example. The folder name will be "Sub2Count (C code)".

**Please Note:** A wrapper code is compiled by Visual Studio. Wrapper searches from "C:\Program Files (x86)" to find all Visual Studio (VS) versions and choose the latest one to compile Wrapper DLL, and this Visual Studio installation folder is registered to the system registry. If Visual Studio is not installed by the default location, Wrapper may not find it; Wrapper pops up a dialog to let the user to specify the folder of VC under the visual studio folder. The wrapper can use any version from Visual Studio 2012, Visual Studio version can be any one of the followings: Industrial, Professional, Community, or Build Tools. Community and Build Tools versions are free downloads from the Microsoft website.

### Step 4: Use Wrapper blocks inside the Embed

Wrapper DLL will be automatically registered under the main menu item "Wrapper Blocks" with the name "My Blocks for \*\*\*", here \*\*\* is the name of Wrapper DLL.

Let's say for this example, we have the subcircuit's name as subBuck.psimsch. Then, the wrapper generates a block for this subcircuit and this can be reached from the main menu inside Embed, as below:



The co-simulation schematic inside the Embed, which utilizes the wrapper block generated from PSIM's subcircuit, is shown below. There is also the PSIMCoupler block which is for co-simulation between PSIM and Embed.



For Target code generation: To make a hardware diagram, one should put hardware device blocks before or after Wrapper blocks, Note that all Wrapper blocks must be used in the hardware diagram. Here we show the sequence to make a hardware diagram for BuckConverter:

 Assume that the hardware target is a F28049 MCU, add a F28x Config block (under Embed //Embedded/F280x/F280x Config... menu) into a new diagram, choose a JTAG type that suites your target board. For example, if the target board is a F280049C LaunchPad, select "TI XDS110-cJTAG USB". You can also change other parameters to suite your target board.

F28x Prop	erties								×
CPU:		F280	049	~					
Enable	Interactive Perip I Out File	heral Mo	ode		lib\setup_F	2837x.out			
CPU Spee	d (MHz):	100		~	Clock Source	ce:	Internal	Oscillator 1	~
Multiple of	Crystal Freq:	10x		~	AUXOSCC	LK=External Crysta	d		~
HSPCLK:	SYSCLK/1	~	100 MHz		LSPCLK:	SYSCLK/4	~	25 MHz	
JTAG con	nection:								
TIXDS110	-cJTAG USB			~					
Control Clk	Src:	32 bit	t timer 2	~	EPWM Inter	rrupt Event:	CTR =	0	~
Control Clk	Prescale:	1		~	Ctrl Clk Cou	nt Mode:	Down		~
Use Cu DLL/VXD Altair Em	stom Linker Cmo Version: bed support for	d File: F280X√1	90 Build 2376						
OK				C	ancel			Help	5

 Buck converter uses an ADC channel (say ADCA1), one should add an Adc Config block (under Embed //Embedded/F280x/ADC/ADC Config... menu) for ADC settings:

SYSCLK:	100 Mhz				A	DC Unit:	A	Y
					Interrupt	on Conv	ersion Start	~
ADCCLK: SYSCLK/	2.5	~	40 Mhz		5	Setup PC	A Gains	
Reference:	Internal 3.3V	~						
Trigger Setup	Src		Trigger		Sample	Clks		
ADCRESULT0:	A1	$\sim$	ePWM1 SOCA	~	15	~		
ADCRESULT1:	A0	~	CPU1 Timer 2	~	15	~		
ADCRESULT2:	A3	~	CPU1 Timer 2	~	7	$\sim$		
ADCRESULT3:	A4/PGA2_OF	~	CPU1 Timer 2	~	7	~		
ADCRESULT4:	A5	~	CPU1 Timer 2	~	7	~		
ADCRESULT5:	VREFLOA	~	CPU1 Timer 2	~	7	~		
ADCRESULT6:	A7	~	CPU1 Timer 2	~	7	~		
ADCRESULT7:	A8/PGA6_OF	~	CPU1 Timer 2	~	7	$\sim$		
ADCRESULT8:	A9	~	CPU1 Timer 2	~	7	~		
ADCRESULT9:	A10/PGA7_OF	~	CPU1 Timer 2	~	7	~		
ADCRESULT10:	PGA1_OUT	~	CPU1 Timer 2	~	7	$\sim$		
ADCRESULT11:	PGA2_OUT	~	CPU1 Timer 2	~	7	~		
ADCRESULT12:	VREFLOA	~	CPU1 Timer 2	~	7	$\sim$		
ADCRESULT13:	PGA5_OUT	~	CPU1 Timer 2	~	7	~		
ADCRESULT14:	PGA6_OUT	~	CPU1 Timer 2	~	7	~		
ADCRESULT15:	PGA6_OUT	~	CPU1 Timer 2	~	7	~		

 Add all SubBuck blocks (only one here) into a new diagram, then add ADC result block (under Embed //Embedded/F280x/ADC/Analog Input menu) before SubBuck block, Since ADC result is a fixed-point number, one should add a convert block to convert the ADC result into float number.  Add PWM1 block (under Embed //Embedded/F280x/PWM/ePWM menu) then set PWM properties as below, then add a convert block set the float number to a fx1.16 fixed-point number:

Rate Scaling:		None	9	~			Cour	nt Me	ode:		Up/D	lown	1	~	
Timer Period:	5000				10kHz	z		1		Chan	ge Pe	riod [	Dynam	nically	
TBCTR=TBP	HS o	n SYNC	l pul	se		тв	PHS	(ph	ase)	t	0				
Change Phas	e Dy	namica	lly			EP	WMS	SYN	ICI pi	in:	GPIO	0			
EPWMSYNCO:	EP	WMSY	NCI		~	EP	WMS	YN	COF	oin:	Unus	ed			~
CMPA Load On:	СТ	R = Zer	o		$\sim$	CM	PBL	oad	d On:		CTR	= Zer	0		$\sim$
Action Qualifier:				CN	DA		Ch								
		Z	u	D	down	_	up	dov	vn	Ρ		_	GPIO	Pin	
EPWMA:		x ~	1	~	0 ~	X	~	х	~	х	~	G	PIO0		V
EPWMB:	1	x ~	x	~	x ~	X	~	x	~	х	~	G	PIO1		~
Deadband:															
Delay Mode:		Disa	bled												×
Polarity:		Inver	tFall	ing	Edge [	Dela	y on	В							
Input Select		DbA	in = {	PWI	MA, Db	8 in	= PW	/MA							
Rising Edge Dela	y (0-1	1023):	0				Fa	Illing	Edg	ge D	elay (O	-1023	3):	0	
Send Start ADC (	CORVE	ersion	Pulso	A	SOCA			C	TR =	PR	2			/1	
Send Start ADC (	Conve	ersion F	ulse	B	SOCB			D	CBE	VT1			~	/1	~
Fault Handling				-					UDL.					/.	
EPWMA output (	on fau	lt	1	Hig	h impe	dand	e			-					
EPWMB output of	on fau	ilt	Ì	Hig	h impe	dand	e			-	D	igital	Comp	are	
Add Enable	Pin (0	value	force	s Fi	ault)										
One Shot TZx Fa	ault S	ource:	[	1	2	2	3		4 [	5	6		DCA		СВ
CBC TZx Fault S	ource	9:	[	1	2	2	]3		4	5	6		DCA		CB
TZ1: GF	001		~	TZ	2 GF	PIO0			~	TZ	3: 0	PIO	)	~	
						0.01					-		TOD		

• The final diagram is as below:



• The hardware system is designed that PWM1 triggers ADC at the end of a PWM period, ADC starts to convert then causes interrupt at the end of conversion. To implement this, one needs to select all blocks from input device block to output device block as below and right mouse click on a selected block to create a compound block.

	TI XDS100	v2 USB				
	AD	C Config				
F280049-ADCARESULT0	► convert	►Taskfin0 SUbBuck_Task TaskfOut0	→ convert	≫%Duty Cycle A(1.18) →%Duty Cycle B(1.18) %Duty Cycle B(1.18)	Alian Left	
					Alian Right	
					Align Stack	
					Align Top	
					Align Inputs	
					Align Outputs	
					Clear	
					Cut	
					Сору	
					Auto Connect	
					Auto Disconnect	
					Create Compound	
					Flip Horizontal	Ctrl+Left

• Input the compound name at the properties page. The diagram is changed as below:



• Ctrl + right mouse click on the compound block, the following properties page shows up:

Compound Properties					
Compound Name					
20KHz					^
Type Ctrl+ENTE	R to enter	r a nev	v line		~
Protection	10	Appea	rance		
Locked Read C	Only	Us	e Bitmap	Select	Image
Password:		Se	t Color		
Do not pause simul	ation when	n Dialo	og pops up		
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> </ul>	ation when contained of <b>RAM</b> round Thre	0.0 0.0	ng pops up nund dialogs		
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> </ul>	ation when contained of RAM round Three Select	0.0 0.0 cad	ag pops up bund dialogs 1 ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> </ul>	ation where contained of RAM round Three Select dler Preen	0.0 ead	ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> <li>Use Local Bounds:</li> </ul>	ation when contained of RAM round Three Select dler Preen Start:	0.0 ottaile ottai ottai ottai ottai ottai ottai ottai ottai ottai ottai ottai ottai ot	1 ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> <li>Use Local Bounds:</li> </ul>	ation where contained of RAM round Three Select dler Preen Start: End:	0.0 0.0 0.0 0.0 0 0 0	ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> <li>Use Local Bounds:</li> <li>Retain State</li> </ul>	ation where contained of RAM round Three Select dler Preen Start: End:	0.0 ead ct 0 0	1 ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Cnabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> <li>Use Local Bounds:</li> <li>Retain State</li> <li>Use Implicit Solver:</li> </ul>	ation wher contained of RAM round Three Selec dler Preen Start: End: Setup S	0.0 ead t 0 0 0 Solver	ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Cneate buttons for of</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> <li>Use Local Bounds:</li> <li>Retain State</li> <li>Use Implicit Solver:</li> <li>Contained Block Count:</li> </ul>	ation when contained of RAM round Three Select dler Preen Start: End: Setup S	0.0 ad at 0 0 Solver	ADCAINT	1.0	
<ul> <li>Do not pause simul</li> <li>Create buttons for of</li> <li>Enabled Execution</li> <li>Copy Flash Function to</li> <li>Local Time Step (sec):</li> <li>Codegen as Backgr</li> <li>Execute on Interrupt:</li> <li>Allow Interrupt Hand</li> <li>Use Local Bounds:</li> <li>Retain State</li> <li>Use Implicit Solver:</li> <li>Contained Block Count:</li> <li>Contained Computational</li> </ul>	ation wher contained of RAM round Three Select dler Preen Start: End: Setup S Blocks:	0.0 ad at 0 0 Solver	ADCAINT	1.0	

• The final diagram is as below:



ADC Config...

#### ► 20KHz/

 One can generate hardware code in Embed main menu //Tools/CodeGen..., a dialog show up as below:

Result File:	BuckConverter	_Hardware.c	
Result Dir:	C:\Altair\Embed	d2022_64\cg	
Target:	F280X		
Subtarget (se	t in target config):	F280049	
Optimization	Level: 0	<u>~</u>	Check for Performance Issues
Use select	ed compound edge p	oins for data exchan	ge (enables embedded debug)
Embed Ma	ps in Code		Add Stack Check Code
	· · ·	Ann	On-Chip RAM Only
Call from F	oreign RTOS/User A	- the	
Call from F	oreign RTOS/ <u>U</u> ser # ock Nesting as Comm	ment	Target ELASH
Call from F	oreign RTOS/ <u>U</u> ser A ick Nesting as Comm emption in Main Dia	ment Igram	Target <u>E</u> LASH
Call from F	oreign RTOS/User A lock Nesting as Comm emption in Main Dia a: 0	nent gram Heap size:	Target ELASH
Call from F	oreign RTOS/ <u>U</u> ser A ick Nesting as Comm emption in Main Dia e: 0 tion Name:	nent gram Heap size: cgMain	Target ELASH

Click "Code Gen" button to generate hardware code, Click "Compile..." button to compile the hardware code, Click "View..." button to show the generated code. Click "Download" button to send the compiled binary code to MCU.

• When clicking on "Download" button, a dialog pops up as below:

Download to F280X	×
Target Execution File:	
C:\Altair\Embed2022_64\cg\BuckConverter_Hardware.out	
This dialog can only download to RAM. Please use Code Con Uniflash to burn your .out file to flash.	nposer or
Quit Coff Info Download	Help

Connect your target board with the specified JTAG, Power on the board, then click "Download" button, the binary code will be sent to the MCU and executed.

This includes the steps on PSIM-Embed wrapper implementation.

# Wrapper's limitation

The PSIM-Embed wrapper is a new enhancement from PSIM v2022.3 release. There are currently some limitations while using Wrapper. When generating a Wrapper code for a PSIM subcircuit, this subcircuit must follow the following rules (including the child subcircuits):

- No power blocks.
- No hardware blocks.
- Not support fixed point type (only floating point type now).
- No bi-directional ports in the current subcircuit and its child subcircuits.
- No PSIM motor control blocks (They uses F28x .obj files, we may compile them to ST or other DSP object file if needed).

- No SimCoder C block.
- No TIDMC blocks.
- Not support IQmath and any blocks use IQmath functions.