**APPLICATION NOTE** 

# Converting Firmware Projects to Colde and IAR Embedded Workbench for ARM

**Power Application Controller**<sup>™</sup>

Marc Sousa Senior Manager, Systems and Firmware



www.active-semi.com Copyright © 2015 Active-Semi, Inc.

No portion of this document may be reproduced or reused in any form without Active-Semi's prior written consent





### TABLE OF CONTENTS

APPLICATION NOTE	1
Table of Contents	2
Overview	3
CooCox Colde	3
IAR Systems Embedded Workbench for ARM	3
Converting to CooCox Colde	5
Creating a Project for the PAC52XX with Colde	5
Configuring Colde for the PAC52XX SDK	7
Copying Application Files to Colde	8
Converting to IAR Systems Embedded Workbench for ARM	10
Creating a Project for the PAC52XX with IAR Embedded Workbench for ARM	10
Copying Files to the IAR Project Directory	11
Configuring the IAR Project for PAC52XX	12
Compiling Functions to Link in RAM	17
About Active-Semi	18



### **OVERVIEW**

Users of the PAC52XX family of controllers have different choices when it comes to development environments. The PAC52XX family of controllers support two main development environments:

- CooCox Colde
- IAR Embedded Workbench for ARM

Users will choose one of these toolsets based on several factors: performance, cost, support and existing skill sets and knowledge.

### CooCox Colde

CooCox's Colde is an Integrated Development Environment (IDE) that supports ARM Cortex MCU based controllers, such as the PAC52XX family. It is a free IDE, based on the Eclipse platform. It supports all the major IDE features and functions such as code editor, compilation tools (which are GCC), debugger and peripheral register viewer.

CooCox also has a SWD emulator used for programming and debugging called CoLinkEx, and CooCox makes the design available for users for this emulator as well. Active-Semi makes a version of this emulator available for customers which offers high-voltage protection of your PC for high-voltage motor and power applications.

As of version 1.7.7, Colde has integrated support for Active-Semi's PAC5210, PAC5220 and PAC5220 and this is the recommended version to download.

The CooCox Colde is available for download on the web at the following URL:

http://www.coocox.org/CooCox\_CoIDE.htm

### IAR Systems Embedded Workbench for ARM

IAR System's Embedded Workbench for ARM supports all available ARM cores, from all vendors. Although there is no specific support for the PAC52XX family, the toolset may be customized to support Active-Semi devices through configuration files from Active-Semi.

IAR Embedded Workbench is a commercial product that must be licensed through IAR Systems. Customers with existing licenses may add support for Active-Semi products under their existing license, as long as they are currently using IAR Embedded Workbench for ARM.

Like CooCox, IAR Embedded Workbench for ARM has a code editor and project manager, compiler and debugger. IAR Embedded Workbench for ARM supports a wide variety of emulators for debugging that are also available for purchase.

More information on IAR Embedded Workbench for ARM can be found at the following URL:

http://www.iar.com/Products/IAR-Embedded-Workbench/ARM/

© 2014 Copyright, Active-Semi International, Inc. - 3 - Rev 1.1 March, 2015





The sections below will illustrate how to convert from other firmware projects to CooCox Colde and IAR Systems Embedded Workbench for ARM.

© 2014 Copyright, Active-Semi International, Inc. - 4 - Rev 1.1 March, 2015



### **CONVERTING TO COOCOX COIDE**

### Creating a Project for the PAC52XX with Colde

Users should begin by downloading version 1.7.7, or later, and installing on their system. For instructions on installation, refer to the online documentation from http://www.coocox.com.<sup>1</sup>

When you launch Colde, on the main screen select "Create a New Project" under "Quick Start" as shown below. You will then see the "New Project" Form.

🕕 CooCox CoIDE					x
File Edit View Project Flash	Debug Search H	łelp			
📑 🗈 🔒 🕮 🚟 🕸 📲	i 🖉   🔿 🖸 🏠	• 🖪 🚳 🛷 🕶 擾 👻 🖗	• <>		
🗖 Components 🛛 👘 🗖	🔞 welcome 🛿			-	· 🛛
Device []	Wolcomo t				<u> </u>
	velcome				
			🕕 New Project		
			Project		
	Comp	onent Repositor	Set project name and path		
		Browse in Repository	Project Name: PAC test proj		Use default path
Project 🕅 🗖 🗖		Component repository contains a			· ·
		peripherals, drivers, examples, and	Project Path: C:\CooCox\CoIDE\works	pace\PAC test proj	
Target					
	Quick	Start			
		Create a New Project			
		Start the wizard that will guide you necessary for creating a new project			
		necessary for creating a new project			
		Open a Project			
		Open an existing CoIDE project.			
			< Back	Next > Finish	Cancel
		User Guide			
		CoIDE User Guide (PDE version) pro	ides details about how		-
			👸 Sign in to CooCox		

Next, type in the name of the project and select your Workspace (or use the default workspace) and press the "Next >" button. You will then see the following form.

<sup>1</sup> Be sure to follow the step to select the tool chain path, or building programs will not work. © 2014 Copyright, Active-Semi International, Inc. -5 - Rev 1.1 March, 2015



Model Create a new project based on your target chip		
Chip	Board	
< Back Next	> Finish	Cancel

Select "Chip" and press the "Next >" button.

You will now see a form where you can select the chip vendor and device. At the bottom of this form, select "Active-Semi", then "PAC52XX" and then the device you are using (for example, the PAC5220).

CH S	New Project nip select the chip for your target pro	oject		2			
	type filter text Atmel Energy Micro	^	Manufacturer: Active-Semi Series: PAC52XX Device: PAC5220 Toolset: ARM				
	Freescale Holtek TI NXP Nuvoton ST Toshiba Spansion Active-Semi PAC5220 PAC5220 PAC5220 PAC5220	4 III	PAC5220 Descriptions: Active-Semi's PAC5220 operates at a maximum speed of 50MHz and features 32 Kbytes of Flash and 8 Kbytes of SRAM. Key Parameters: • 4*24-bit timer (TMER) • 28°GPIO(General Purpose Input Output) • 3*0TIF-PGA (differential programmable gain amplifiers) • 4*PGA (programmable gain amplifiers) • 10*Comparators • 2*0AC(10-bit and 8-bit) • 11*ADC(10-bit 1)				
	< Back Next > Finish Cancel						

After selecting the device, click the "Finish" button and the project will be created.

You can examine the configuration options by right-clicking on the project and selecting "Configuration". You will then see the "Configuration" tab and you can examine the project parameters. Below shows the "Link" tab which shows linker parameters for the PAC52XX.



	Compile	Link	Output	User	Debugger	Download				
.ink										
Option	s						Linked Lil	braries		
🗸 Use	Memory I	.ayout fr	om Memo	ory Wind	low					Add
V Disc	ard unuse	d sectio	ns							
V Don	't use the	standard	l system st	tartup fil	les					Remove
1 Shares	Net									
Library	INOT U	se C Libr	ary 🔻							
Memor	y Areas –									
	only Men	nory Are	а				Read/wri	te Memory Area		
Read/		Star	t		Size		on-chip	Start		Size
Read/ on-ch	ip			0.00			IRAM1	0x20000000	0x000	02000
Read/ on-ch IROM	ip 1 0x00	000000		UXU	0008000					
Read/ on-ch IROM IROM	1 0x00 2	000000			0008000		IRAM2			
Read/ on-ch IROM IROM	1 0x00 2 2	000000 sh	ebua in R		0008000		IRAM2			
Read/ on-ch IROM IROM	ip 1 0x00 2 pug in Fla	000000 sh () D	ebug in R		0008000		IRAM2			
Read/ on-ch IROM: IROM: O Deb Scatter	ip 1 0x00 2 Dug in Flas File	000000 sh 🔘 D	ebug in R	AM	0008000		IRAM2			

The project will be created with just one file (main.c) and will now build. The project directory should look like shown below.

CoIDE 🛛	<ul> <li>workspace</li> <li>PAC test proj</li> </ul>	👻 🐓 Sear	rch PAC test proj	٩
rary 🔻	Share with 🔻 New folder			• 🔳 🔞
^	Name	Date modified	Туре	Size
	main.c	9/26/2014 1:27 PM	C File	1 KB
	PAC test proj.cogui	9/26/2014 1:27 PM	COGUI File	18 KB
	PAC test proj.comarker	9/26/2014 1:27 PM	COMARKER	1 KB
E	PAC test proj.coproj	9/26/2014 1:27 PM	CoIDE Project	4 KB

Press the "F7" key to build this simple project to verify that the default project was correctly created.

### Configuring Colde for the PAC52XX SDK

Next, Colde will need to download and configure support for the PAC52XX SDK. To start this process, select the "Repository" view and click the "Refresh Basic Component List in the upper right hand corner.

		SIIO
ble	CooCox	
ble	CooCox	
ble	<u>CooCox</u>	
ble	CooCox	

© 2014 Copyright, Active-Semi International, Inc. - 7 - Rev 1.1 March, 2015

No portion of this document may be reproduced or reused in any form without Active-Semi's prior written consent



**Power Application Controller** 

Colde will now download the components needed for Active-Semi from CooCox. When complete, the repository will now show the "CMSIS Core and BOOT" and "LIB" components for Active-Semi as shown below.

	Step	3 Select Basic Compo	nents [Active-Semi / PAC5220]		<del>8</del> S
	Ξ	COMMON			
		Semihosting	Implementation of Semihosting GetChar/SendChar	Available	CooCox
		Retarget printf	Implementation of printf(), sprintf() to reduce memory footprint	Available	CooCox
		C Library	Implement the minimal functionality required to allow newlib to link	Available	CooCox
		M0 Cmsis Core	CMSIS Core For Cortex M0 V 3.01	Available	CooCox
		METABuffer	Applies a FIFO or LIFO behavior to any buffer of any kind of elements.	Download	Kairos (Author not verified)
		TimeOut	Object that implement a timeout mechanism based on signal/callback events. Usefu	Download	Kairos (Author not verified)
-	Ξ	BOOT			
_		CMSIS Core and BOOT	CMSIS Core and BOOT for Active-Semi PAC5XXX	Download	CooCox
-	Ξ	PERIPHERAL.COX			
		CoX_Interface	The standard interface of CoX.CoX Peripheral Library is the definition of a group of	Available	CooCox
		RTOS			
		CooCox OS	CooCox OS V1.1.6 for Cortex M0 kernel	Available	CooCox
		FRT_FreeRTOS	FreeRTOS 6.0.4	Update	K.A. (Author not verified)
		FRT_RTOS_layer	A higher level layer for FreeRTOS use for FRT.	Update	K.A. (Author not verified)
	•	PERIPHERAL.ACTIVE-SE	MI		
		LIB	Library for Active-Semi	Download	CooCox
	۲.	1 1	m		
	Manu	facturers Chips Peripher	als Drivers Others		
			🔒 Sign in to CooCox		

The user should then check both of these boxes, in order to build programs for Active-Semi's PAC52XX family. When you check each of these boxes, Colde will ask if you want to download the component. Click "OK", and then after these downloads, this form should show these components as "installed", as shown below.

		Semihosting	Implementation of Semihosting GetChar/SendChar	Avail
		Retarget printf	Implementation of printf(), sprintf() to reduce memory footprint	Avail
		C Library	Implement the minimal functionality required to allow newlib to link	Avail
		M0 Cmsis Core	CMSIS Core For Cortex M0 V 3.01	Avail
		METABuffer	Applies a FIFO or LIFO behavior to any buffer of any kind of elements.	Dow
		TimeOut	Object that implement a timeout mechanism based on signal/callback events. Usefu	Dow
		BOOT		
Ξ	<u>ک</u> آ	CMSIS Core and BOOT	CMSIS Core and BOOT for Active-Semi PAC5XXX	Avail
est proj	-	PERIPHERAL.COX		
t proj		CoX_Interface	The standard interface of CoX.CoX Peripheral Library is the definition of a group of	Avail
n.c		RTOS		
		CooCox OS	CooCox OS V1.1.6 for Cortex M0 kernel	Avail
		FRT_FreeRTOS	FreeRTOS 6.0.4	Upda
		FRT_RTOS_layer	A higher level layer for FreeRTOS use for FRT.	Upda
	8	PERIPHERAL.ACTIVE-SEP	л	
		LIB	Library for Active-Semi	Avail

This is the CMSIS boot and PAC52XX SDK support needed to build programs for the PAC52XX family. To be sure that these are properly installed, build the project by pressing the "F7" key.

Now, the entire toolset configuration is all set for porting the firmware from another IDE.

### **Copying Application Files to Colde**

The user should now copy all the application firmware to the project directory. In this example, I have copied over a project into the project directory from a BLDC motor controller. The steps I have followed are below.

© 2014 Copyright, Active-Semi International, Inc. - 8 - Rev 1.1 March, 2015



Power Application Controller

Next, copy application firmware files and directories. Do NOT copy any CMSIS, SDK or other project files from the old installation or old IDE. The project file should look similar to this.

🍌 cmsis_boot	9/26/2014 1:40 PM	File folder	
🍌 cmsis_core	9/26/2014 1:40 PM	File folder	
🌡 cmsis_lib	9/26/2014 1:40 PM	File folder	
🗼 ISR	9/26/2014 1:44 PM	File folder	
🍌 PAC test proj	9/26/2014 1:30 PM	File folder	
application.c	8/28/2014 3:03 PM	C File	8 KB
application.h	8/28/2014 3:56 PM	H File	10 KB
fix16.c	5/28/2013 9:57 AM	C File	20 KB
fix16.h	9/25/2013 11:51 AM	H File	7 KB
init.c	8/28/2014 3:55 PM	C File	13 KB
int64.h	5/28/2013 9:56 AM	H File	4 KB
main.c	8/25/2014 2:26 PM	C File	4 KB
PAC test proj.cogui	9/26/2014 1:27 PM	COGUI File	18 KB
PAC test proj.comarker	9/26/2014 1:27 PM	COMARKER	1 KE
🕕 PAC test proj.coproj	9/26/2014 1:40 PM	CoIDE Project	9 KE
📄 pid.c	8/27/2014 9:45 AM	C File	4 KE
📄 pid.h	11/13/2013 1:49 PM	H File	3 KE
Sensored_BLDC_PID.cogui	8/28/2014 3:57 PM	COGUI File	29 KE
Sensored_BLDC_PID.comar	8/28/2014 3:57 PM	COMARKER	3 KB
Sensored_BLDC_PID.come	4/10/2014 2:42 PM	COMEMGUI	1 KB
🕕 Sensored_BLDC_PID.coproj	8/28/2014 4:12 PM	CoIDE Project	10 KB
uart.c	8/28/2014 3:03 PM	C File	11 KB
version.h	10/3/2013 4:30 PM	H File	1 KB

Now all the files are in the project directory, needed for the build.

The user must now add the files to the project, as follows.

- For files, user must right-click on the project name (in the Project frame), and select "Add Files", then select the files they wish to add
- If the user wishes to add a group (like a directory) to the project, then they should right-click on the project name, and select "Add Group". In the example below, the group "ISR" has been added by the user. The groups "cmsis\_boot", "cmsis\_core" and "cmsis\_lib" have been added by Colde, when the Active-Semi SDK has been configured in the project through the repository
  - If the user has added a group, to add files to the group the user right-clicks the group, then selects "Add Files" to add files to this group.

Finally, the project may be built by pressing the "F7" key.



### CONVERTING TO IAR SYSTEMS EMBEDDED WORKBENCH FOR ARM

#### Creating a Project for the PAC52XX with IAR Embedded Workbench for ARM

Users should begin by downloading and installing the version of IAR Embedded Workbench for ARM on their system, according to the installation instructions.

IAR Embedded Workbench does not have integrated support for the PAC52XX, so the user needs to perform some one-time configuration of the toolset before projects may be created. These instructions are available in the SDK for IAR on the Active-Semi website.

Once the PAC52XX support is installed into the IAR Embedded Workbench for ARM, and new project for the PAC52XX can be created.

The first step is to create a new project by selecting "Project" and then "New Project..." from the main menu. A dialog box will appear, and the user should select an empty project.

Create New Project			23
Tool chain:	ARM		•
Project templates:			
Empty project • asm • C++ • C • DLIB (C, C++ v • DLIB (C, Exten	vith exceptions and R1 ded Embedded C++)	ΓTI)	
Description:			
Creates an empty p	oject.		
		ОК	Cancel

After this, the new project has been created in IAR Embedded Workbench for ARM as shown below.



For each new project, and IAR "workspace" must be created. The workspace is a file that references a set of different projects so that the user can have a set of projects to manage at the same time in the IDE.

© 2014 Copyright, Active-Semi International, Inc. - 10 - Rev 1.1 March, 2015



If the user has not already selected a workspace, then they should save one by selecting "Save Workspace" from the "File" menu.

After the project and workspace have been saved, the project directory should contain the following files.

	Spectra and project fields a		X
🕞 🕘 🗸 🗼 🕨 iar_new_proj 🕨	<ul> <li>✓ ✓ ✓ Search iar_new_pri</li> </ul>	oj	٩
Organize 🔻 😭 Open 🛛 Ir	nclude in library 🔻 Share with 💌 New folder	••• •	
revortes     Desktop     Desktop     Recent Places     Google Drive     Google Drive     Grf     ownCloudAs     WVPC EVK (A6)     WVPC EVK (A1)     WPC EVK (A11)     Documents     Documents     Music     Pictures     Pictures     Pictures     Pictures	my_wkspc.eww psc52x_new_proj_jar.dep pac52x_new_proj_jar.ewp		
settings Date modif File folder	ied: 9/26/2014 3:19 PM		

The project directory will now contain the IAR project files (\*.dep, \*.ewp, \*.ewd) and optionally the workspace file (\*.eww).

### **Copying Files to the IAR Project Directory**

Once the project has been completed, then files may be added to the project directory.

Unlike Colde, the SDK is <u>not</u> integrated into IAR Embedded Workbench, so the IAR SDK must be downloaded from the Active-Semi website, and then copied into the project directory. In the example below, the SDK files are put directly into the project directory, and a main.c file is created to call some of the SDK functions.

	Spattle, see, pro, or United .		
🕒 🗢 🕌 🕨 iar_new_proj 🕨		✓ 4 Search iar_new	_proj 🔎
Organize 🔻 Include in library 🔻	Share with 🔻 New folder		
Favorites     Dektop     Recent Places     Soogle Drive     Grand Drive	iii pac52∞, dk muinc pac52∞, new_proj_lar.dep pac52∞, new_proj_lar.exp	ii settings ii my_wkspc.eww pacS2a_new_proj_lar.ewd	
□ Libraries         □ Documents         □ Music         □ Pictures         □ Videos         □ 7 items			

Once these files are copied into the project directory, the files must be added to the project.

To add files, the user right-clicks on the project (in the Workspace tab) and selects "Add", then "Add Files...". The user may then select the files to add to the project.

To group files into directories in the project, the user adds a group by right clicking the project name, then selecting "Add", then "Add Group....". The group name is added and then users may add files to this group by right-clicking on the group and selecting "Add" and then "Add Files...".

© 2014 Copyright, Active-Semi International, Inc. - 11 - Rev 1.1 March, 2015



**NOTE**: Be sure to NOT copy any project files or any files that begin with "." to this directory, that come from the previous toolset.

For projects that use the PAC52XX SDK, all the SDK files must be added to the project.

Once the files have all been added to the project, the IDE will look something like this.

🔀 my_wkspc - IAR Embedded Workbench IDE	
File Edit View Project Simulator Tools Window Help	
D 🚅 🖬 🞒 🏯 🕺 🖻 💼 🔁 🗠	- 🗸 🏷 🐂 🖳 🖻 🗢 🗢 🏟 🕼 🔤 🔤 🖗 🧖
Workspace	main.c f0 • ×
Debug	
Files	Ra Ba void main (void)
🗆 🗇 pac52xx_new_proj_iar - Debug	✓ L <sup>1</sup>
	E
pac52xx_new_prol_iar	<
Ready	L

#### **Configuring the IAR Project for PAC52XX**

Once all files have been added to the IAR project, then the project must be configured for the PAC52XX device.

The user must first select the PAC52XX device to be used. To do this, the user must right-click the project in the Workspace tab on the left, and select "Options...". The option dialog box will come up and the user should select the "General Options" tab. The user should then select the Active-Semi PAC52XX device as shown below.

Satenory [								
Seneral Options								
C/C++ Compiler								
Assembler Output Converter	Target	Output	Library	Configurati	on l	Library Ontions	MISBA-C:20	0.4.>
Custom Build	-	a a de la de				actually optimized		
Build Actions								
Linker	Proc	essor var	riant					
Debugger	0	0.00	Cortex-I	V0		r		
Simulator		2010						
Angel	0	)evice	Active-S	Semi PAC5	2XX		<b>1</b>	
CMSIS DAP								
GDB Server								
IAR ROM-monitor	E							
I-jet/JTAGjet		an moue					_	
J-Link/J-Trace	0	ittle			None		-	
TI Stellaris	0	ig						
Macraigor		) BE32						
PE micro	6	BE8						
RDI								
ST-LINK								
Ihrd-Party Driver								

The user should then go the "Library Configuration" tab and select the "Use CMSIS" check-box. CMSIS is used for the PAC SDK and some of the other ARM peripheral header files for using the ARM Cortex peripherals.

© 2014 Copyright, Active-Semi International, Inc. - 12 - Rev 1.1 March, 2015



X Options for node "pac52xx\_new\_proj\_iar" Category: C/C++ Compiler Assembler Target Output Library Configuration Library Options MISRA-C:200 4 Output Converter Custom Build Library: Description: **Build Actions** Use the normal configuration of the C/C++ Linker Normal runtime library. No locale interface, C locale, no Debugger file descriptor support, no multibytes in printf and Simulator scanf, and no hex floats in strtod. Angel CMSIS DAP Configuration file GDB Server \$TOOLKIT\_DIR\$\INC\c\DLib\_Config\_Normal.h IAR ROM-monitor Enable thread support in library I-iet/JTAGiet -CMSIS J-Link/J-Trace - Library low-level interface implementation TI Stellaris None stdout/stdem-Use CMSIS Macraigor Semihosted Via semihosting DSP library PE micro IAR breakpoint Via SWO RDI ST-LINK Third-Party Driver XDS100/200/ICDI OK Cancel

Next, the user should select the "C/C++ Compiler" category on the left and the "Optimizations" tab to select the level of compiler optimization the application needs. In most cases, the higher levels are used (High/Balanced) to provide adequate optimization.

XDS100/200/ICDI	Category: General Options C(C++ Compler Assembler Output Converter Cutput Conve	Multi-file Compilation          Discard Unused         Language 2         Code         Level         None         Low         Medium         Igh         Balanced         No size con	d Publics Optimizations Enal V V V S Astraints	Factory Settings           Output         List         Preprocessor         * *           bled transformations:         *         *         *           common subexpression elimination         Loop unrolling         *         *           Function inlining         Code motion         *         *         *           Static clustering         Isstate clustering         *         *         *           Instruction scheduling         Vectorization         *         *         *
-----------------	--	---	--	---

The user must now configure the include paths and symbols used by the application, by clicking on the "Preprocessor" tab on this form.

In the "Additional include directories: (one per line)" text box, the user should add the include paths for all header files needed to build the program. For example, the SDK include paths for the header files should be added.

The user must also define the symbol "IAR" on this form. When complete, this form should look as follows below.

© 2014 Copyright, Active-Semi International, Inc. - 13 - Rev 1.1 March, 2015



X

**Power Application Controller** 

Options for node "pac52xx\_new\_proj\_iar"

General Options       Multi-file Compilation         C/C++ Compiler       Discard Unused Publics         Assembler       Discard Unused Publics         Output Converter       Language 1 Language 2 Code Optimizations Output List f         Custom Build       Build Actions         Linker       Ignore standard include directories         Debugger       Ignore standard include directories         Simulator       Additional include directories: (one per line)         Angel       SPROJ_DIRS\pac5xxx_sdk\pac5xxx_drivers         GDB Server       IStellaris         J-Link/J)-Trace       Preinclude file:         TI Stellaris       Defined symbols: (one per line)         RDI       IAR         Third-Party Driver       XDS 100/200/ICDI	Category:	Factory Settings
Third-Party Driver XDS 100/200/ICDI	General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI STJ IM/	
	Third-Party Driver XDS 100/200/ICDI	Generate #line directives

The IAR toolset tends to use more stack space than CooCox Colde. To change the stack space for the project, the user should select the "Linker" list item then check the "Override default" check box, and enter the new stack size in the tool window as shown below.

Category: General Options	Factory Settings
C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel	Config Library Input Optimizations Advanced Output List
CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK	Edit         Configuration file symbol defin         Linker configuration file editor         Vector Table         Memory Regions         Stack/Heap Sizes         CSTACK         0
Third-Party Driver XDS 100/200/ICDI	Save Cancel

In this example, the user has changed the stack size to 0x180.

© 2014 Copyright, Active-Semi International, Inc. - 14 -

Rev 1.1 March, 2015



The project should now be fully configured so that it can be built. It may be built by pressing the "F7" button.

To configure the debugger for use with the I-Jet, the user should select the "Debugger" item from the list and select the "I-jet/JTAGjet" selection from the pull-down menu:

ages Extra Options Multicore Plugins
ages Extra Options Multicore Plugins       Image: Bun to       Image: main
ages Extra Options Multicore Plugins
Run to     main )
v main
)
)
)
/
e
CONFIG\debugger\ActiveSemi\pac52gr.ddf
0

After making this selection, the user should select the "Download" tab to select the FLASH loader.

Next, the user should select the "Download" tab, to configure the adapter:

Factory Settings
Setup       Download       Images       Extra Options       Multicore       Plugins         Attach to running target       Merfy download         Suppress download       Suppress download         Use flash loader(s)       Qvenide default       board file         \$TOOLKIT_DIR\$\config\flashloader\ActiveSemi\pac          Edt       Edt

The tools will now use the FLASH loader for the PAC52XX to download the programs to the target.

Next, the user should instruct the debugger to turn off power to the target, and select SWD as the communications interface for the debugger.

© 2014 Copyright, Active-Semi International, Inc. - 15 - Rev 1.1 March, 2015



Power Application Controller

To disable the adapter powering the target the user should select the "I-jet/JTAGjet" from the pull-down menu and uncheck the "From the probe" check-box from the target power group box:

Category:	Factory Setting:
General Options	
C/C++ Compiler	
Assembler	
Output Converter	etup JTAG/SWD SWO Breakpoints
Custom Build	Reset
Build Actions	
Linker	System (default)
Debugger	Duration 200 Dalay
Simulator	Duration: 300 ms Delay after: 200 ms
Angel	
CMSIS DAP	larget power
GDB Server	Erom the probe
IAR ROM-monitor	Leave on after debugging
I-jet/JTAGjet	Switch off after dobugging
J-Link/J-Trace	Switch on alter debugging
TI Stellaris	
Macraigor	Log communication
PE micro	
RDI	\$PROJ_DIR\$\cspycomm.log
ST-LINK	
Third-Party Driver	
XDS 100/200/TCDT	

Finally, the user should select SWD by changing to the "JTAG/SWD" tab and selecting the "SWD" radio button from the "Interface" group box:

Category:		Factory Settings
Seneral Options		
C/C++ Compiler		
Assembler	Catura ITAG/SW	ID CIMO Developation
Output Converter	Setup 017(0/300	5WO Breakpoints
Custom Build	Probe config	Probe configuration file
Linker	Auto	Override default
Debugger	From file	
Simulator	Curriliant	
Angel	Depiron	CFU: Select
CMSIS DAP	Interface	Explicit probe configuration
GDB Server	ITAG	Multi-target debug system
IAR ROM-monitor	© 0110	Target number (TAP or Multidron ID): 0
I-jet/JTAGjet	0 <u>5</u> WD	
J-Link/J-Trace		Target with multiple CPUs
TI Stellaris		CPU number on target: 0
Macraigor	JTAG/SWD speed	JTAG scan chain contains non-ABM devices
PEINICIO	Auto detect 👻	
ST-I INK		Preceding bits:
Third-Party Driver		
XDS100/200/TCDT		

© 2014 Copyright, Active-Semi International, Inc. - 16 - Rev 1.1 March, 2015

No portion of this document may be reproduced or reused in any form without Active-Semi's prior written consent



### **COMPILING FUNCTIONS TO LINK IN RAM**

To optimize performance in the PAC architecture, time-critical functions may execute in RAM, instead of FLASH memory. Different sets of tools have different ways to force the linker to mark a function to be linked into RAM (the .data section) instead of FLASH (the .code section).

If using the PAC52XX SDK, the user may mark functions to be complied in RAM as follows:

```
PAC5XXX_RAMLINK void my_function(void)
{
   /* Function */
}
```

The MACRO "PAC5XXX\_RAMLINK" is defined in pac5xxx.h. This MACRO places a token before the function, which allows the compiler and linker to identify it to place into RAM.

If the user is not using this MACRO, each supported toolset has a different way to link functions into RAM.

To link a function into RAM using CooCox Colde, the user needs to add the token "\_\_attribute\_\_((section(".data"))) before the function, as follows:

```
__attribute__((section(``.data"))) void my_function(void) {
    /* Function */
}
```

This method also works for any GNU GCC compiler implementation.

To link a function into RAM using IAR Embedded Workbench for ARM, the user needs to add the keyword "\_\_ramfunc" before the function, as follows:

```
__ramfunc__ void my_function(void) {
    /* Function */
}
```

Using these techniques, the linker will place these functions into RAM, in the .data section according to the linker memory map.



### **ABOUT ACTIVE-SEMI**

Active-Semi, Inc. headquartered in Dallas, TX is a leading innovative semiconductor company with proven power management, analog and mixed-signal products for end-applications that require power conversion (AC/DC, DC/DC, DC/AC, PFC, etc.), motor drivers and control and LED drivers and control along with ARM microcontroller for system development.

Active-Semi's latest family of Power Application Controller (PAC)<sup>™</sup> ICs offer high-level of integration with 32-bit ARM Cortex M0, along with configurable power management peripherals, configurable analog front-end with high-precision, high-speed data converters, single-ended and differential PGAs, integrated low-voltage and high-voltage gate drives. PAC IC offers unprecedented flexibility and ease in the systems design of various end-applications such as Wireless Power Transmitters, Motor drives, UPS, Solar Inverters and LED lighting, etc. that require a microcontroller, power conversion, analog sensing, high-voltage gate drives, open-drain outputs, analog & digital general purpose IO, as well as support for wired and wireless communication. More information and samples can be obtained from <a href="http://www.active-semi.com">http://www.active-semi.com</a> or by emailing marketing@active-semi.com

Active-Semi shipped its 1 Billionth IC in 2012, and has over 120 in patents awarded and pending approval.

#### LEGAL INFORMATION & DISCLAIMER

Copyright © 2012-2013 Active-Semi, Inc. All rights reserved. All information provided in this document is subject to legal disclaimers.

Active-Semi reserves the right to modify its products, circuitry or product specifications without notice. Active-Semi products are not intended, designed, warranted or authorized for use as critical components in life-support, life-critical or safety-critical devices, systems, or equipment, nor in applications where failure or malfunction of any Active-Semi product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Active-Semi accepts no liability for inclusion and/or use of its products in such equipment or applications. Active-Semi does not assume any liability arising out of the use of any product, circuit, or any information described in this document. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of Active-Semi or others. Active-Semi assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein. Customers should evaluate each product using Active-Semi products. Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. All products are responsible for the use of sale supplied at the time of order acknowledgment. Exportation of any Active-Semi products may be subject to export control laws.

Active-Semi<sup>™</sup>, Active-Semi logo, Solutions for Sustainability<sup>™</sup>, Power Application Controller<sup>™</sup>, Micro Application Controller<sup>™</sup>, Multi-Mode Power Manager<sup>™</sup>, Configurable Analog Front End<sup>™</sup>, and Application Specific Power Drivers<sup>™</sup> are trademarks of Active-Semi, I. ARM® is a registered trademark and Cortex<sup>™</sup> is a trademark of ARM Limited. All referenced brands and trademarks are the property of their respective owners.

© 2014 Copyright, Active-Semi International, Inc. - 18 -

Rev 1.1 March, 2015