EmbeddedCraft crafting of intelligent systems

IAR WORKBENCH FOR 8051 PART3

CONTENTS

Creating Project Profiling Interrupt Simulation



Advance debugging

- 1. Profiling
- 2. Code coverage
- 3. Breakpoints
- 4. Tracing
- 5. Simulation of interrupts



C- SPY debugger







Must Remember 1/2



Output file should have debug information

Options for node "serie	s" X
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Chipcon ROM-Monitor Analog Devices Silabs Simulator	Factory Settings Language Code Optimizations Output List Preprocessor D • • Make library module Object module name Image: Setting info Image: Setting info Image: Setting info Image: Setting info Image: Setting info Image: Setting info Image: Setting info Image: Setting info
	OK Cancel

Must Remember 2/2



Linker Format: debugging information for debugging

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Chipcon ROM-Monitor Analog Devices Silabs Simulator	Output Extra Output #define Diagr Output file Override default series.d51 With runtime control modules With I/O emulation modul Buffered terminal output Allow C-SPY-specific extra Other Output format: intel-ex Format variant: None Module-local symbols: Include a	Factory Settings
---	---	------------------

Profiling



- Display time consumed in each function
- So that user can check, which function is taking more CPU time and identify bottle neck problem,
- This will help to make code fast and

* 🖸 🖬 🖬 G O

	Function	Calls	Flat Tim	Flat Time (%)	Accumu	Accumulated Time (%)	
	Outside main	0	0		0		
	low_level_init	0	0		0		
	do_foreground_process	1	83		296		
	get_fib	1	26		26		
	init_fib	0	0		0		
	main	0	2		298		
	next_counter	1	10		10		
	put_fib	1	169		177		
Ē	putchar	2	8		8		
rofi							
ΩI							

Profiling/Code Coverage/Trace



- These are to be loaded as plugins
- So enable them from
- General options > Debugger > Plugins

Options for node "serie	s''			x
Category:			Factory Settings	1
General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Chipcon BOM Monitor	Setup Extra Select plugins OBTI BIO Profiling Trace	Options Plugins s to load: erage DS		
Analog Devices Silabs Simulator	Description:	Enables ORTLRTOS support in the debu	ugger. ed Workbench 4.05	
	Originator:	IAR Systems		
	Version:	1.0		
		OK	Cancel	

Profiling/Code Coverage/Trace



- These are to be loaded as plugins
- So enable them from
- General options > Debugger > Plugins

Options for node "serie	≥s"		×
Category:			Factory Settings
General Options		·	
C/C++ Compiler	Setup Extra (Options Plugins	
Assembler Custom Build	Select plugins	s to load:	
Build Actions	Code Cove	erage	
Linker)S	
Debugger	I I P rofiling		
Third-Party Driver	Trace		
Chipcon	✓Stack		
RUM-Monitor			
Silabo			
Simulator	Description:	Enables URITIRIUS support in the debu	gger.
	Location:	D:\Program Files\IAR Systems\Embedde	d Workbench 4.05
	Originator:	IAR Systems	
	Version:	1.0	
		1	
	L		
		ОК	Cancel

Profiling/Code Coverage/Trace



- These are to be loaded as plugins
- So enable them from
- General options > Debugger > Plugins

Options for node "serie	s"	×
Category:		Factory Settings
General Options	·	
C/C++ Compiler	Setup Extra	Options Plugins
Assembler Custom Build	Select plugin:	s to load:
Build Actions	Code Cov	erage
Linker		JS
Debugger Third-Partu Driver	Proming	
Chipcon	Stack	
ROM-Monitor		
Analog Devices		
Silabs Simulator	Description:	Enables ORTI RTOS support in the debugger.
	Location:	D:\Program Files\IAR Systems\Embedded Workbench 4.05
	Originator:	IAR Systems
	Version:	10
	V CISION.	
		·
		OK Cancel



Open project series >> rebuild >> open Debugger

💥 IAR Embedded Workbench	IDE
File Edit View Project Debug	Simulator Tools Window Help
File Edit View Project Debug Image: I	Simulator Tools Window Help Simulator Tools Window Help Ities tutor 41 void main(void) 42 { 43 call_count=0; 44 45 init_fib(); 46 47 while(call_count < MAX_FIB) 48 { 49 do_foreground_process(); 51 } 52]
y Status Bar genasm series try ↓ ↓	00016D 65 D0 00016F 33 000170 40 EC ↓ 000172 22

Profiling: Enabling profile	EmbeddedCraft crafting of intelligent systems
Enable Profile	
* Function Calls Flat Tim Flat Time (%) Accumu	Accumulated Time (%)
Ready	NUM //

EmbeddedCraft Profiling: Enabling profile crafting of intelligent systems Auto Refresh × 0 2 1 T F 1 C 0 Flat Time (%) Function Calls Flat Tim... Accumu... Accumulated Time (%) Outside main 0 0 Π _low_level_init 0 0 n do_foreground_process 0 0 Ο get_fib 0 0 Ū init_fib 0 0 n 0 0 main Ο put_fib putchar 0 0 0 next_counter 0 0 0 Π Π Π Ready Ln 52, Col 1 NUM



×	0 d t f f d C O					
	Function	Calls	Flat Time (cycles)	Flat Time (%)	Accumulated Time (cycles)	Accumulated Time (%)
	Outside main	0	126		126	
	low_level_init	1	0		0	
	do_foreground_process	10	830		3348	
	get_fib	10	260		260	
	init_fib	1	386		386	
	main	1	162		3896	
2	next_counter	10	100		100	
늰	put_fib	10	2062		2158	
Å,	putchar	24	96		96	
Rea	eady Ln 43, Col 3 NUM					

Profiling: Detail of any function 1/2



Select function and click on detail

π.		- L	· · · · · · ·		
				- ™®₁	
	15.2				

	Function	Calls	Flat Time (cycles)	Flat Time (%)	Accumulated Time (cycles)	Accumulated Time (%)		
	Outside main	0	126	3.13	126	3.13		
	low_level_init	1	0	0.00	0	0.00		
	do_foreground_process	10	830	20.63	3348	83.20		
	get_fib	10	260	6.46	260	6.46		
	init_fib	1	386	9.59	386	9.59		
	main	1	162	4.03	3896	96.82		
	next_counter	10	100	2.49	100	2.49		
E	put_fib	10	2062	51.24	2158	53.63		
ð L	putchar	24	96	2.39	96	2.39		



×							
	Function	Calls	Flat Time (cycles)	Flat Time (%)	Accumulated Time (cycles	s) Accumulated Time (%)	
	Outside main	0	126	3.13	126	3.13	
	low_level_init	1	0	0.00	0	0.00	
	do_foreground_process	10	830	20.63	3348	83.20	
	get_fib	10	260	6.46	260	6.46	
	_init_fib	1	386	9.59	386	9.59	_
	main	1	162	4.03	3896	96.82	
l g	next_counter	10	100	2.49	100	2.49	
١Ę	put_tib	10	2062	51.24	2158	53.63	
Pro	putchar	24	96	2.39	96	2.39	
×	Function: main						
	Flat time 162 cycles	, Accu	mulated time 3896	cycles.			
	Callers:						
	Total: 1						
	Count Eurotion						
5	1 Unknown calle	er(s)					
Detai	Callees:						
ling I	Count Function						
Profi	4						Þ
Re	ady					NUM	/



Display the execution of code, step by step,

can trace the value of any variable, after each line of execution

×	er 👬 🕽	< 🗈 🐆	
	PC	call_count	
	0x15C	1	
	0x16D	1	
	0x15C	2	
	0×16D	2	
	0x15C	3	
	0x16D	3	
	UX15C	4	<u> </u>
e e	- ·		
Ĕ	Expressions	Output	
Rea	ady		NUM //





Open series project >> compile >> debug

XIAR Embedded Workbench	IDE		- U ×
File Edit View Project Debug	g Simulator Tools Window Help		
Messages Messages Workspace Source Browser		▼ >> >> > >> >> >> >> >> >> >> >> >> >> >> >> >>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Source Browser Workspace Debug Files Disassembly Memory Register Watch Locals Auto Live Watch Quick Watch Call Stack Terminal I/O Code Coverage Profiling Trace Stack Toolbars Status Bar	<pre>s tutor</pre>	Disassembly Go to Co Co Co Co Co Co Co Co Co C	★ de LJ. ▲ NO NO SJ. LJ.L. LJ.A NO NO SJ. LJ.L. LJ.S NO SJ. LJ.L. NO NO SJ. SJ. NO NO SJ. SJ. SJ. SJ. SJ. SJ. SJ. SJ. SJ. SJ.
Overview Interrul		1	► OM
Open the Trace window		Ln 40, Col 3 NUM	



Trace: setting 3 Two Tabs



× 🛯 💥 🗙 🖹 🔖	
PC	
g Europeaniene (O. e. e.)	
Ready	

Enter the value of variable, want to examine





	Enable trace	
Enable Trace		
× # # X • ×		1
Expressions Output		



	XIAR Embedded Workbench IDE	
	File Edit View Project Debug Simulator Tools Window Help	
		🗾 🏷 🎾 🌆 🖬 🏷 🖉
Reset	(둘)▮불ᆋॾॾॾॾ	
	Workspace × utilities tutor ×	Disassembly ×
	Debug 19 oid next_counter(void)	Go to Code
	Files 😤 📴 20	000000 02 00 4E LJA
	□ tuto ✓ 21 call_count += 1;	?ROM_MONITOR_NOPS:
	∰ tut 23	exit:
	24.¥	000004 00 NO
	25 Increase the 'call c	000005 00 NO. 000006 80 FB SJ:
	27 /	?cmain:
	28 oid do_foreground_proce	000008 12 00 57 LC.
		00000B B9 00 03 CJ
	× 🐼 🗙 🗈 🐂	
	PC	
	0x3	
		- 1
	Expressions Output	
	Ready	Ln 40, Col 3 NUM



	🔀 IAR Embedded Workbench IDE	
	File Edit View Project Debug Simulator Tools Window Help	
		🔄 🏷 🍾 🧏 📓 🖻 🍫 🕼 🌬
Execute Code	<u></u> □ □ □ □ □ □ □ □ □ □	
stop by stop	Workspace	Disassembly
sieh ny sieh	Debug	Go to 💽 🔽 Code
	Files 62 B	000000 02 00 4E LJ .
	21 call_count += 1;	PC_EXIT: PROM MONITOR NOPS:
	22 	exit:
		000004 00 NO
	25 Increase the 'call_c	000005 00 NO: 000006 80 FB SI
	26 Get and print the as	?cmain:
	28 oid do_foreground_proce	low_level_init_call: 000008 12 00 57 LC.
	29	00000B B9 00 03 CJ
	Overview Interrul () fo () int. fih:	
	PC	
	0x3	
		_
	Ready	Ln 40, Col 3 NUM



	💥 IAR Embedded Workbench IDE	
	File Edit View Project Debug Simulator Tools Window Help	
	D 🖆 🖵 📮 😂 👗 🖻 💼 🗠 က 🖂 💽 🖻	-] 🍾 🍾 🔁 🔝 📼 📣 🎼 🛙
Execute Code	〒 	
sten hv sten	Workspace × utilities tutor • × Disasse	embly ×
Step by Step	Debug I9 oid next_counter(void) Go to	Code
	Files 😤 📴 20	00000 02 00 4E LJ
	21 call_count += 1; 70 22 22	OM_MONITOR_NOPS:
		it: 00003 00 NO
	24 ^{·★}	00004 00 NO
	25 Increase the 'call c U	00005 00 NO.
	27 / ?c	main:
	28 oid do_foreground_proce 0	00008 12 00 57 LC
	29 30 ungigned int file	0000B B9 00 03 CJ
	× 🐼 💥 X 🗈 🐂	
]
		=
		-1
	8	
	Expressions Output	
	Ready In 40,	Col 3 NUM



5 8 2 2 2 2	ラジジ 😕 🙊		
Workspace ×	utilities tutor	isassembly	×
Debug 💌	38 Main program.	Go to 📃 💽 Code	•
Files 👫 🛤	39 Prints the Fibonacci	000152 78 21	MO
🗖 🖻 tuto 🗸	40/	000154 78 00	
⊞ 📓 tut	42	000157 76 00	MO.
	43 call_count=0;	000159 12 01 73	LC.
□	44	while(call count <)	MAI
	45 init_fib();	00015E C3	CL:
	\Rightarrow 40 while (call count < M	00015F E6	MO SII
	48 {	000162 08	IN _1
Overview Interru	fol I do foreground proces	000163 E6	MO'Ľ
× 🛹 🐺 🗙 🗈 🎙	4		
PC			
0x3			
0x3			
0x152			
0x159			
I I I I I I I I I I I I I I I I I I I			키
Ready		NUM	



	💥 IAR Embedded Workbench IDE			
	File Edit View Project Debug Simulati	or Tools Window Help		
) CH		
	5 5 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	0		
	Workspace × utilities between		Disassembly	×
		Main nuoman 🗖	Go to	Code
	50 mg 30	Prints the Fibonacci	000152 78 21	MO
	Files Cr. Dig 40 /		000154 76 00	MO
	41 oi	d main(void)	000156 08 000157 76 00	MO
	42 ⊞ 🗟 uti 43 c	all count=0:	<u>init fib();</u> 000159_12_01_73	тс
	44		while(call count	< MAI
	45 i	nit_fib();	➡ 00015C 78 21 00015E C3	MO CL
		hile(_call_count_/ M	00015F E6	MO
	48 (000160 94 0A 000162 08	SU. IN
	Overview Interrul 4 > 49	do foreground proces	000163 E6	MO.
Select expression tab				
	🍈 🚧 🔐 🗙 🗈 🐂			
	Expression	Output Format		
	call_count			
	Expressions Output			
	Ready		NL	
			, j	111







1	IAR En	nbedde	ed Workl	ench I	DE								- D ×
File	Edit	View	Project	Debug	Simula	ator Tool	s Window	Help					
) 🖻	8	18	ХB	C.	ကဂျ				• > >	: 🔀 🖟		🖻 🎒 🛙
Îz		12	æ & ;	y 25	2	8							
Wo	rkspace		×	utilitie	s tuto	or l		+ ×	c Disa	ssembly			×
D	ebug		•					-	- Go	o to			ode
F	iles	2			45	init_fi	b();	-	' [000156	08		
	陷 tu	to	 I 	-	46 47	while(call co	unt < M		<u>init f</u>	ib();		MO
	- 🕀 🔝	tut			48	{				000159 while(12 01 'call	73 count	LC.
Ht	-te) 📳 -te) 💼	uti			49	do_fo	reground	_proces	 	00015C	78 21		MO
I.		····			50	}				00015E 00015F	E6		MO
					52					000160 000162	94 OA 08		SU: TN:
Ŀ										000163	E6		MO
Ľ-		. Inter					_	-	1.	000164 0/0166	94 00 A2 D2		MO
15	verviev			fo]	•					
×[es.	* X	(🗈	4									
	PC		C	all_cou	int								
	0x150	2	1										
	0x160	5 C	2										
	0x16	Ž	2										
	10x15 10x16	ŏ	3										
	0x15	0	4										⊡
Trace	Expres	sions	Output										
Rea	dy											NUM	



Shows which part of the code is executed





To open Code Coverage window

XIAR Em	bedded Workbench	IDE
File Edit	View Project Debug	g Simulator Tools Window Help
File Edit	bedded Workbench View Project Debug Messages Workspace Source Browser Breakpoints Disassembly Memory Register Watch Locals Auto Live Watch Quick Watch Call Stack Terminal I/O Code Coverage Profiling Trace Stack	<pre>IDE Simulator Tools Window Help ies in the second of the second of</pre>
	Toolbars Status Bar	<pre>41 void main(void) 42 { 43 call_count=0; 44 45 init_fib(); 46</pre>







To see code coverage execute the code step by step or execute completely





Details displayed in code coverage window





Details displayed in code coverage window





Details displayed in code coverage window







To generate a report right click in the window





Save the file (file is in the text format)

	Save Code Cove	age Data			<u>? ×</u>
Overview gena Log Tue Nov 28 1 Tue Series 3	Save in: Save in: History Desktop My Documents	My Documents My Computer My Network Place codecoverage	ces	← 🛍 📸 💷	
	My Network P	File name: Save as type:	<mark>codecoverage</mark> Text Files (*.txt)	•	Save Cancel

http://www.embeddedcraft.org

