北京中科昊芯科技有限公司

Haawking-IDE 用户使用手册

V1. 6. 0

北京中科吴芯科技有限公司

2021 年 12 月

版本号	修改日期	修改人	修改内容
V0. 0. 1	2020. 09. 09	齐如士	初稿
V0. 0. 6	2020. 11. 2	吴军宁	IDE 升级
V0. 1. 0	2021.02.01	吴军宁	IDE 升级
V0. 6. 0	2021.06.01	吴军宁	IDE 升级
V1. 2. 0	2021.08.02	吴军宁	重大版本升级
V1. 2. 1	2021.08.07	吴军宁	增强稳定性
V1. 3. 0	2021.09.01	王思腾	重大版本升级
V1. 4. 0	2021.09.16	王思腾	重大版本升级
V1. 5. 0	2021.11.12	王思腾	增强稳定性
V1. 5. 1	2021.11.19	王思腾	增强稳定性
V1.6.0	2021.12.25	王思腾	增强稳定性

版本更新记录

注1:版本号注意区分重大更新和部分修改,版本更新需审批后决定。

注 2: 前三位为 IDE 版本号, _分割为文档更新版本号。

V1.2.0版本(仅支持量产版芯片)

更新 IQMath 函数库,补全乘法以及类型转换的相关函数

更新驱动库

默认使用 stdlib 库文件

更新 027 和 034 外设寄存器 json 文件

V1.2.1版本

IDE 升级,包括:1.提升 28027 擦除及烧写的主频为 12MHz,及修改 Flash 对应主频的配置参数;2. 修复 Erase 函数 Bug;增加时钟校准功能;28027增加 ENPIE 的定义,将系统时钟设为 120M,修改 delay 函数名字,1abs 警告;3. 28034 增加 ENPIE 的定义,将系统时 1 钟设为 120M,修改 IQmath 的表格,在 bootrom 中的定义,修改 delay 函数名字,1abs 警告;4.修改 28335 的 1d 文件的 fPU 表 格

V1.3.0版本

- 1. 增加了 segger 实时运行库
- 2. 驱动库更新记录:
 - (1) 修正文件名规范为 DSP2802x
 - (2) 增加对__interrupt 的支持
 - . IQMath 函数库更新记录:
 - (1) 增加了 intof 函数
 - (2) 增加对 IQcosTable 的支持
 - (3) 提高 mpy 函数的运行速度
- 4. OpenOCD 版本升级为 VO. 1. 6, 更新内容为:

(1) 支持 DSC28027 主频为 3MHz、12MH 下的擦写(在配置 DivSel 成功的 情况下,以 12MHz 主频的速度擦写 Flash,否则以 3MHz 主频擦写 Flash)

(2) 缩短了读 FLASH 状态标志位超时时间为 10 秒

(3) 增加芯片加密状态弹窗

V1.4.0版本

- 1. Driver 更新记录:
 - (1) 支持调试时对内部时钟晶振校准
- 2. IDE 功能更新记录:
 - (1) 提供更便捷的新建工程方式
 - (2) 实时刷新模块支持全局变量的实时刷新

V1.5.0版本

- 1. 更新驱动库
- 2. IDE 功能更新记录:
 - (1) 提供新版界面的实时刷新视图, 增加稳定性
 - (2) 增强了 Haawking Project 创建工程方式的稳定性
- 3. OpenOCD 版本升级为 0.2.0

V1.5.1版本

1. 更新驱动库

V1.6.0版本

- 1. IDE 功能更新记录:
 - (1) 增加 Debug Without Download 功能
 - (2) 增加直接烧入程序功能
 - (3) 支持生成静态库文件
 - (4) 支持切换 RAM/FLASH 后,无需执行 Clean Project
- 2. Openocd 版本升级为 0.2.1
- 3. 更新驱动库

目录

1 Haawki	ing IDE 介绍	8
1.1	Haawking IDE 目录结构	8
1.2	Haawking IDE 常见问题	.11
	1.2.1 Haawking IDE 不支持导入友商工程	.11
	1.2.2 调试器	.11
	1.2.3 调试时遇到的问题及解决方法	.11
	1.2.4 如何添加源文件和头文件目录	.14
	1.2.5 怎样获得反汇编文件	.17
	1.2.6 如何切换已有工程 FLASH 和 RAM	.19
	1.2.7 头文件、函数、变量无法跳转到声明位置	.20
	1.2.8 Haawking IDE 不支持在任务栏中启动	.21
	1.2.9 IDE 路径包含中文字符或者空格	.22
1.3	HaawkingIDE 工程目录介绍	.23
1.4	Map 文件简要说明	.24
1.5	HaawkingIDE 暂未支持的功能	.25
	1.5.1 工程重命名	.25
	1.5.2 实时波形显示	.25
2 Haawk	ing IDE 使用	.26
2.1	打开 Haawking IDE	.26
2.2	新建工程	.27
2.3	~	.32
	2.3.1 整个工程编译	.32
	2.3.2 单个文件编译	.34
2.4	如何生成库文件	.35
	2.4.1 前置工作	.35
	2.4.2 生成静态库	.37
	2.4.3 使用静态库	.39

I	中		科	旲	芯	文	档	
		2.5	编译优化	介绍				40
			2.5.1 -0.	1 级别优体	Ł			
			2.5.2 -02	2 级别优体	Ł			
		2.6	生成 bin ⁵	格式文件		•••••		
		2.7	下载功能					43
			2.7.1 前	置工作				43
			2.7.2 使	用 IDE 下载	战程序			45
		2.8	导入工程	<u>.</u>				48
		2.9	导出工程	<u>.</u>				50
			2.9.1 导	出压缩包格	各式工程			50
			2.9.2 导	出文件夹格	各式工程			51
		2.10) 使用 Re	build Proje	ect 功能			53
		2.1	1 Debug 🗄	目录下的文	件介绍			54
		2.1	2 用 Spik	e 模拟器句	方真程序			54
	3 诉	閉试						57
		3.1	启动调试			••••••		57
		3.2	调试模式	工具栏介约	召			59
		3.3	断点的使	用及断点礼	观图介绍			60
			3.3.1 如	何添加断点	<u>,</u>	•••••		60
			3.3.2 如	何删除断点	<u>,</u>			60
		X	3.3.3 使	用 Breakpo	oints 视图管	管理断点		60
		\mathbf{v}	3.3.4 Bre	akpoints 礼	见图功能按	钮介绍		61
			3.3.5 FLA	、SH 工程使	更用断点的消	主意事项		
		3.4	如何使用	Restart 功	I能复位芯J	т Т		62
			3.4.1 FLA	、SH 工程使	可用 Restart	功能		62
		3.5	如何查看	变量值		•••••		64
		3.6	如何查看	外设寄存着	器的值			65

3.7 使用 Memort 视图查看存储器中的值......67

3.7.1 如何查看指定地址处的数据	67
3.7.2 如何导出 Memory 视图中的数据	67
3.8 如何进入汇编调试	68
3.9 使用 Debug Without Download 功能	69
3.9.1 前置工作	69
3.9.2 启动 Debug Without Download	71
3.9.3 Debug Without Download 功能注意事项	72
3.10 使用实时刷新功能	72
3.10.1 实时刷新视图使用注意事项	72
3.10.2 实时刷新视图按钮介绍	73
3.10.3 向实时刷新视图添加变量	73
3.10.4 删除实时刷新视图中的变量	75
3.10.5 切换变量输出格式	75
3.10.6 切换刷新间隔	76
3.10.7 数据存在变化时的显示效果	76
3.10.8 导出实时刷新视图中的数据	77
3.10.9 实时刷新视图存在的问题	78
3.11 如何加密解密	79
3.11.1 加密操作	79
3.11.2 解密操作	80
4 审核	83

1 Haawking IDE 介绍

核心功能	IDE 1.2.0	IDE 1.3.1	IDE 1.4.0	IDE 1.5.1	IDE 1.6.0
实时刷新变量功能					
(手册 3.10 小节)	-	-	~	\checkmark	\checkmark
调试时复位芯片功能					
(手册 3.4 小节)	-	—	~	\checkmark	\checkmark
新版的创建工程方式					\sim
(手册 2.2 小节)	_	_	~		~
Rebuild Project 功能					
(手册 2.10 小节)	_	_	_	~	~
Debug Without Download 功			\boldsymbol{X}	X	
能	_	_ >		_	~
(手册 3.9 小节)					·
生成静态库功能					
(手册 2.4 小节)	_	_	_	_	\checkmark
直接烧入程序功能					
(手册 2.7 小节)	-	—	-	—	~

1.1 Haawking IDE 目录结构

用户可以从昊芯官方网站(http://haawking.com/zyxz)下载到最新版本的 Haawking IDE 软件和相关的驱动库,首次使用,请下载完整包 (Haawking-IDE-win64-V1.6.0.exe),包括编译器、JRE环境、HX-Link以及 OpenOCD 等工具。

_	IDE 下载 ●
	下载链接 版本:Haawking-IDE-win64-V1.5.1.exe / 1.25GB 阿里云做地址:http://www.aliyundive.com/v/wwSgBCMLnGH
	历史版本: 阿里阿盘
V1.5.1 版本更新说明	
 更新服功库 IDE 功能更新记录: 1) 提供新版界面的读时间新规图,增加稳 (2) 增强了 Haawking Project 创建工程方式 3. OpenOCD 版本升级为 0.2.0 	性の現実性

解压之后,将会得到下图所示的文件,包括 eclipse 目录、开发工具 haawking-tools 目录、Hawking-IDE 用户手册、Haawking-IDE-win64-V1.6.0. exe 文件。

名称 ^	修改日	期		类型	大小
eclipse	2021/	12/24	15:48	文件夹	
📙 haawking-tools	2021/	12/11	17:20	文件夹	
🔢 Haawking-IDE-win64-V1.6.0	2021/	3/13 1	5:25	应用程序	3,337 KE
open_ide	2021/	8/30 1	5:19	Windows 批处理	2 KE
open_ide_spike	2021/	8/19 1	5:03	Windows 批处理	1 KB
◎ 北京中科吴芯科技有限公司-Haawking-ID	2021/	12/27	17:55	WPS PDF 文档	8,484 KE

注意:用户需记住 Haawking IDE 的解压路径,说明示例的解压路径为 <u>D:\Haawking-IDE-win64-V1.6.0</u>;后面很多操作均依赖该路径。

☆: Haawking IDE 的解压路径中不能包含中文字符及空格,否则无法正常 工作。建议解压在根目录!

Haawking IDE 的开发工具都集中在 haawking-tools 目录,包括编译器、链接器、调试器等。该目录的文件内容如下图:

6

wking-IDE-win64-V1.5.0 >	haawking-tools »	~	ひ / 搜索"	naawking-tools"
名称	~ 修	改日期	类型	大小
compiler	20	21/10/9 11:19	文件夹	
dwex_wst	20	21/11/1 10:44	文件夹	
📙 haawking-devices-libs	20	21/8/14 13:28	文件夹	
<mark>]</mark> jdk-11.0.8	20	21/8/14 13:28	文件夹	
📙 jlink	20	21/8/14 13:28	文件夹	
openocd	20	21/8/14 13:28	文件夹	
📙 spike	20	21/8/14 13:28	文件夹	
📧 haawking_ide_config	20	20/11/5 18:06	应用程序	112 KB

Haawking IDE 的编译器基于 LLVM 12.0.0 开发, 链接器 1d 和调试器 gdb 则 使用了 RISC-V GCC 10.2.0 官方发布版本。OpenOCD 是根据最新源码编译的官方 版本。

开发工具后续存在升级的可能,使用者可以根据需要选择对某一开发工具单 独进行升级。在升级的时候,只需替换相关路径下的文件即可。

☆☆☆: 使用 Haawking IDE 之前,请先阅读第一章的 2、3、4、5 小节。

1.2 Haawking IDE 常见问题

使用过程中,如果遇到FAQ 未列的情况,可以 <u>https://gitee.com/haawking/haawking-tools/issues</u>下面<u>按照格式要求</u>,提 交问题反馈,开发人员会尽快给予回复。

Haawking IDE V0.0.6-IQmathLib-Bug

```
    #123538 ③待办的 JunningWu 拥有者 创建于 2天前
    IDE名称: Haawking-IDE-Eclipse-CDT.win32.x86_64_V0.0.6
    IDE版本: V0.0.6
    运行环境: FPGA-ZYNQ-MZ7035FD,主频10MHz
    问题描述: 如下
    原因分析: 库文件中的函数调用存在bug
    解决办法:
    建议:
    勾选Bootrom Table选项,需要这样写程序,才可以运行出正确结果。
    a = _IQ2in(998);
    而写成下面的形式,则运行结果错误。
    a = _IQ29sin(998);
    在不勾选Bootrom Table选项时, a = _IQ29sin(998); 可以得到正确的结果。
```

1.2.1 Haawking IDE 不支持导入友商工程

Haawking IDE 不支持导入、编译及调试友商工程。

1.2.2 调试器

我们始终建议客户选择 HX-Link 进行调试。

1.2.3 调试时遇到的问题及解决方法

1.2.3.1 设备连接错误之一

当出现下面错误信息,说明 HX-Link 调试器与开发板未正确连接。请检查 JTAG 线序或因为排线过长导致的信号不稳定。建议使用中科昊芯提供的 14-PIN 排线或连接线不长于 14cm。

中	科	旲	芯	文	档

Console 🛛 Memory Reprogress Reproblems Securables Reproduces Console <terminated> HX_DSC28027_FLASH_120MHZ_APPOINTED_ADDR_TEST_V_0_0_5-debug-oocd [GDB Op Error: JTAG scan chain interrogation failed: all ones Error: Check JTAG interface, timings, target power, etc. Error: Trying to use configured scan chain anyway... Error: riscv.cpu: IR capture error; saw 0x1f not 0x01 Warn : Bypassing JTAG setup events due to errors Error: Unsupported DTM version: 15 Error: Target not examined yet

Error: Unsupported DTM version: 15

1.2.3.2 设备连接错误之二

当出现下面错误信息,说明芯片未上电,需要检查供电或检查芯片是否按正 确方向电装。



1.2.3.3 设备烧写错误

当出现下面错误信息,说明芯片在烧写 Flash 时,烧写校验失败。建议用户 在烧写前检查芯片的供电或芯片周边是否有电磁干扰,或降低".cfg"配置文件 中的调试频率的参数,调试频率的参数最低为 500,默认单位为"KHz"。排查 问题后,可复位芯片重新烧写。

Flash Sector 0/63 Loading... Sector 0/63 Verify Succeed Flash Sector 0/63 Loading... Started by GNU MCU Eclipse Error: Load Sector Failed... Error: write data is 0x7d, read data is 0x0 Error: Load Sector Failed... Error: write data is 0x15, read data is 0x0 Error: write data is 0x23, read data is 0x23 Error: write data is 0x4f, read data is 0x4f Error: write data is 0x4f, read data is 0x4f Error: write data is 0x4f, read data is 0x4f Error: write data is 0x6, read data is 0x60 Error: write data is 0x6, read data is 0x6e Error: write data is 0x82, read data is 0x82

".cfg" 文件的位置在

"/haawking-drivers/haawking-dsc28027-board/board/"下,修改配置参数见下图:

💼 main.c	C f2802x	sysctrl.c	h f2802x_e
1 adapt	er speed	5000	
2 #reset	t config tre	t only	
3 #adap	oter srst pu	Ise widt	h 50
4		0.00	

1.2.3.4 芯片复位导致与仿真器断开连接

当出现下面错误信息,说明芯片复位导致与仿真器断开连接。是由于程序中 未关闭看门狗导致,或因程序配置问题引起芯片复位。因此需要用户检查程序代 码。



1.2.3.5 断点设置过多

在 RAM 工程中, 断点的数量是任意的。

在 Flash 工程中,用户最多添加两个断点。

注意: Debug 模式下, Flash 工程中再加入第三个断点, 或进入 Debug 模式 之前就断点数量就已经超过了三个, 那么控制台就会报如下错误:

中	科	旲	芯	文	档	
		📮 Console 🛛	Memory 1	🖁 Registers 🛛 😽 Pro	ogress 🖹 Problems 🕻	
		HX_DSC28027_F	LASH_120MHZ_A	PPOINTED_ADDR_	TEST_V_0_0_5-debug-oo	
		Flash Sector	0/63 Loading	g		
		Sector 0/63	Verify Succee	ed		
		Flash Sector	0/63 Loading	g		
		Sector 0/63	Verify Succee	ed		
		Flash Sector	2/63 Loading	g		
		Sector 2/63	Verify Succee	ed		
		Flash Sector	63/63 Loadir	ng		
		Sector 63/63	Verify Succe	eed		
		Flash Sector	63/63 Loadir	ng		
		Sector 63/63	Verify Succe	eed		
		Flash Load F	inished!			
		Started by G	NU MCU Ecli	ipse		
		Error: Could	n't find an av	ailable hardwa	ire trigger.	
		Error: can't a	add breakpoi	nt: resource no	ot available	

导致芯片无法再继续运行,需要点击关闭按键 B 关掉进程,再移除所有断点后 再重新进入 Debug 模式。

1.2.3.6 警告提示但不影响调试

以下的调试警告信息提示,只要不是一直在循环提示,是不影响芯片的正常 调试的。



1.2.4 如何添加源文件和头文件目录

在实际的使用中,除了芯片的驱动库和"BSP"之外,可能需要添加自己的 源文件目录和头文件目录,除了可以将源文件和头文件放在"src"目录之外, 还可以手动添加。

1.2.4.1 添加头文件目录

下面以 hello_haawking 工程下的 test_include 目录作为演示添加头文件的例 子,如下图所示。

中	科	旲	芯	文	档	
יע 🗲 איγ י	ello_haawking	9				
> 6	Includes		1 20027 1	i.		
> 2	haawking-d src	rivers/haawkin	g-dsc28027-b	oard		
> 6	≻ Debug ≻ haawking-d	rivers				
>@	∍ lib ∍ test include	_				
I I I I I I I I I I I I I I I I I I I	⊨ <u>⊫</u> test_inclu	de.h				
		angaaunen				
	实	列头文件目录	及头文件			

右键点击项目名称,再点击"Properties",打开项目属性选项卡。"C/C++ Build→Settings→Tool Settings→GNU RISC-V Cross C Compiler→Includes"。

Properties for hello_haawk	ing		- 🗆 X
type filter text	Settings		\$
> Resource Builders V C/C++ Build Build Variables	Configuration: Debug [Active]		✓ Manage Configurations
Environment Logging Tool Chain Editor > C/C++ General > MCU Project Natures Project References Run/Debug Settings Task Tags > Validation WikiText	 Tool Settings Tool Settings Tool Settings Conclusins Devices Conclusins Devices Conclusins Devices Conclusing Devices Conclusing Devices Settings Marings Marings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Sental History Cross C Compiler Warnings Miscellaneous Sental History Cross C Linker General Ubraries Miscellaneous Sental NISC-V Cross Create Flash Image General Sental NISC-V Cross Print Size General 	Container Settings Build Steps Build Artifact Binary Parsers Container Settings Include paths (-i) Chaawking-drivers/haawking-dsc28027-board/comman* Chaawking-drivers/haawking-dsc28027-board/f2802x_common/include* Chaawking-drivers/haawking-dsc28027-board/f2802x_common/include* Include system paths (-isystem) Include files (-include) Include files (-include)	Parsers 원 원 원 전 원 ⓒ 환 원 원 전 원 환 원 전 원 전 환 원 원 전 원
			Restore Defaults Apply
(?)			Apply and Close Cancel

在"Include paths (-I)"中,点击"Add..."按钮,会弹出选择头文件 目录的面板,由于例子中的头文件目录位于 hello_haawking 项目中,因此可以 使用相对路径的方式,即输入"../test_include",然后点击 OK 按钮。

注:添加头文件时,使用"../"路径就代表项目根路径。

	科	旲	芯	文	档	
Include	paths (-I)					월 18 8 61 64
/haawl	king-drivers/haa	wking-dsc28027-b	oard/common	aders/include"		
/haaw	king-drivers/haa	wking-dsc28027-b	oard/f2802x_co	ommon/include"		
Ac State	dd directory patł	1			×	
Dire	ectory:					
l/te	est_include					· 10 10 10 10 10 10 10 10 10 10 10 10 10
	1 ./代表当	前工程的路径	Astest inclu			
		Cancel	Worker	ace File svs	tem	
nclude	e files (-include)					· 월 월 월 월
1	~ 느 스 시 그		A			
于上海	添加完头又	件目求后,	会理出	卜力的对记	ī框,点击"	Yes"按钮即可。
k s	ettings	10				×
-	Channel					a la la Dana da la
(?	to rebui	s made will n ild it now?	ot be refie	cted in the ir	idex until it is r	rebuilt. Do you wish
				×		
E		1				
	Remember	r my decision				
1	Remember		Ø		X	
1]Remember 该按钮	r my decision 用于记住选择	译		Yes	No

1.2.4.2 添加源文件目录

下面以 hello_haawking 工程下的 test_src 目录作为演示添加头文件的例子, 如下图所示。

✔ 📂 hello_haawking	
> 🐉 Binaries	
> 🗊 Includes	
> 😕 haawking-drivers/haawl	king-dsc28027-board
> 🔁 src	
✓ 2 test_src	
> c test_include.c	示例源又件目录

对于源文件目录,如果不想放在"src"目录下,可以通过项目属性选项卡,

"C/C++ General → Paths and Symbols → Source Location",点击"Add Folder..."按钮,找到并添加自己的源文件目录后,点击 Apply and Close 按钮即可。



1.2.5 怎样获得反汇编文件

V1.5.1版本的 IDE,默认会输出可执行文件的反汇编文件,在 Debug 目录下, 文件已".asm"和".s"后缀结尾,分别由"LLVM"和"GCC"反汇编工具生成, 其中".asm"支持中科吴芯自定义指令助记符显示;二者指令编码一样。

全称 修改日期 英型 大小 src src Src Src 本 packs D:Users\abbedipse-workspace\hello_haawking\Debug\hello_haawking.asm - Notepad+++ 文件D 编辑() 微索() 微离() 语言() 设置① T具() 余(M) 运行() 操件() 密门() hello_haawking.asm 日目の haawking.asm hello_haawking.asm 日目の haawking.asm hello_haawking.asm 日日の haawking.asm joigetts.mk 日日の haawking.elf: file format ELF32-riscv assembly of section .text.init: 1 for 00012000 _start: 1 a 12000: 81 40 mv fa, zero joigetts.mk 1 joigetts.mk 5 <	cipse workspace > hello_haawkin	g / Debug			* 0	12.51 0	ebug		~
src D:Uuserstabledipse-workspace(hello_haawking\Debugthello_haawking.asm - Notepad++ 文件氏 编辑E 搜索S 视图公 编码N 语言() 设置() 工具() 余M 运行K 编件P 管() D.hello_haawking.eff hello_haawking.map makefile D objects.mk 1 2 hello_haawking.eff hello_haawking.map makefile D objects.mk 1 2 hello_haawking.elf : file format ELF32-riscv 1 1 2 hello_haawking.elf : file format ELF32-riscv 1 2 hello_haawking.elf : file format ELF32-riscv 1 1 2 hello_haawking.elf : file format ELF32-riscv 1 1 2 hello_haawking.elf : file format ELF32-riscv 1 12002 : 01 41 1 12002 : 01 41 1 12002 : 01 41 1 12004 : 81 42 1 12006 : 81 42 1 12006 : 81 42 1 12006 : 81 42 1 12006 : 81 42 1 12006 : 81 43 1 <t< th=""><th>_ 名称 ^</th><th>修</th><th>改日期</th><th>类型</th><th>大小</th><th></th><th></th><th></th><th></th></t<>	_ 名称 ^	修	改日期	类型	大小				
Type of the state Type of the state <t< td=""><td>srr</td><td>D:\Us</td><td>ers\ab\eclipse-works</td><td>oace\hello haawking</td><td>\Debug\hello h</td><td>aawking.</td><td>asm - N</td><td>otepad++</td><td></td></t<>	srr	D:\Us	ers\ab\eclipse-works	oace\hello haawking	\Debug\hello h	aawking.	asm - N	otepad++	
Image: Section seccinc seccin section section section section section s	xpacks	文(件(F)	编辑(F) 搜索(S) 视图		い没習の工具	∃(O) ≠	:(M) i≅	行(R) 插件(P)	窗口(
<pre>hello_haawking.eff hello_haawking.hex hello_haawking.map makefile bojects.mk</pre>	hello haawking.asm			hhipei	# ha @ @			1 🔳 🖉 🕅	
<pre>bello_haawking.hex bello_haawking.map makefile bojects.mk bello_haawking.map bojects.mk bisassembly of section .text.init: bisassembly of section .text</pre>	hello haawking.elf								
<pre>hello_haawking.map makefile model bello_haawking.elf: file format ELF32-riscv hello_haawking.elf: file fo</pre>	hello haawking.hex	plugin	.xml🖾 🔚 template.xr	n1 🖾 📕 dsc28027-xs	vd/json🖬 🔚 pl	ugin.xml	🖾 📃 p.	lugin. properti	estis 📕
makefile a bojects.mk a bojects.mk b column 2000 section sources.mk b column 2000 start: a 12002 column 2004 81 41 mv start column 2004 81 42 mv tot column 2004 81 42 mv tot column 2004 81 43 mv tot column 2004 81 43 mv tot column 2004 81 43 mv <t< td=""><td>hello haawking.map</td><td>1</td><td>alla baanbing</td><td>-16. 6:1- 6-</td><td>mat ELESS</td><td></td><td></td><td></td><td></td></t<>	hello haawking.map	1	alla baanbing	-16. 6:1- 6-	mat ELESS				
B Objects.mk 4 Sources.mk 5 Disassembly of section .text.init: 6 7 00012000 _start: 8 12000: 81 40 mv ra, zero 9 12002: 01 41 mv gp, zero 10 12004: 81 41 mv gp, zero 11 12006: 01 42 mv t0, zero 12 12008: 81 42 mv t0, zero 13 12002: 01 43 mv t1, zero 14 1200e: 01 44 mv s0, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a3, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero	makefile	3	lerro_naawking.	eil: ille Ioi	mat ELF32-	TISCA			
5 Disassembly of section .text.init: 6 00012000 _start: 8 12000: 01 40 mv ra, zero 9 12002: 01 41 mv pp, zero 10 12004: 81 41 mv pp, zero 11 12006: 01 42 mv tp, zero 12 12008: 81 42 mv tp, zero 13 12002: 01 43 mv t1, zero 14 1200e: 01 44 mv s0, zero 15 1200e: 01 44 mv s0, zero 16 12010: 81 44 mv a0, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a2, zero 19 12016: 01 46 mv a3, zero 20 12018: 81 46 mv a4, zero	D objects.mk	4							
6 7 00012000 _start: 8 12000: 81 40 mv ra, zero 9 12002: 01 41 mv gp, zero 10 12004: 81 41 mv gp, zero 11 12006: 01 42 mv t0, zero 12 12006: 01 42 mv t0, zero 13 12002: 81 43 mv t1, zero 14 12002: 81 43 mv s0, zero 15 12002: 01 44 mv s0, zero 16 12012: 81 44 mv s1, zero 18 12014: 81 45 mv a0, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 12018: 81 47 mv a4, zero	Sources.mk	5 1	isassembly of	section .text	.init:				
7 00012000_start: 8 12000: 81 40 mv ra, zero 9 12002: 01 41 mv gp, zero 10 12004: 81 41 mv gp, zero 11 12006: 01 42 mv th, zero 12 12008: 81 42 mv th, zero 13 12006: 01 43 mv t1, zero 14 12006: 01 44 mv s0, zero 15 12006: 01 44 mv s1, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		6							
8 12000: 81 40 mv ra, zero 9 12002: 01 41 mv rp, zero 10 12004: 81 41 mv rp, zero 11 12006: 01 42 mv tp, zero 12 12006: 81 42 mv t0, zero 13 12006: 81 43 mv t1, zero 14 1200e: 01 44 mv s0, zero 15 1200e: 01 44 mv s1, zero 16 12010: 81 44 mv a0, zero 17 12012: 01 45 mv a0, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 12018: 01 47 mv a4, zero		7 0	00012000 _start						
9 12002: 01 41 mv sp, zero 10 12004: 81 41 mv gp, zero 11 12006: 01 42 mv tp, zero 12 12008: 81 42 mv t0, zero 13 1200a: 01 43 mv t1, zero 14 1200a: 81 43 mv t2, zero 15 1200e: 01 44 mv s0, zero 16 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 12018: 01 47 mv a4, zero		8	12000: 81 40)		mv	ra, :	zero	
10 12004: 81 41 mv gp, zero 11 12006: 01 42 mv tp, zero 12 12008: 81 42 mv tp, zero 13 12002: 01 43 mv t1, zero 14 12002: 81 43 mv t2, zero 15 12002: 01 44 mv s0, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 12014: 01 47 mv a4, zero		9	12002: 01 41			mv	sp, a	zero	
11 12006: 01 42 mv tp, zero 12 12008: 81 42 mv t0, zero 13 1200a: 01 43 mv t1, zero 14 1200e: 01 44 mv s0, zero 15 1200e: 01 44 mv s1, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a2, zero 19 12016: 01 46 mv a3, zero 20 12018: 81 46 mv a4, zero		10	12004: 81 41			mv	gp, a	zero	
12 12008: 81 42 mv t0, zero 13 1200a: 01 43 mv t1, zero 14 1200c: 81 43 mv t2, zero 15 1200e: 01 44 mv s0, zero 16 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		11	12006: 01 42	2		mv	tp, a	zero	
13 1200a: 01 43 mv t1, zero 14 1200c: 81 43 mv t2, zero 15 1200e: 01 44 mv s0, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		12	12008: 81 42			mv	t0, :	zero	
14 1200c: 81 43 mv t2, zero 15 1200e: 01 44 mv s0, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		13	1200a: 01 43	3		mv	t1, :	zero	
15 1200e: 01 44 mv s0, zero 16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		14	1200c: 81 43	3		mν	t2, :	zero	
16 12010: 81 44 mv s1, zero 17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		15	1200e: 01 44			mν	s0, 1	zero	P
17 12012: 01 45 mv a0, zero 18 12014: 81 45 mv a1, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		16	12010: 81 44			mν	s1, :	zero	
18 12014: 81 45 mv al, zero 19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		17	12012: 01 45	5		mν	a0, 2	zero	
19 12016: 01 46 mv a2, zero 20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		18	12014: 81 45	5		mv	a1, 2	zero	
20 12018: 81 46 mv a3, zero 21 1201a: 01 47 mv a4, zero		19	12016: 01 46	5		mv	a2, 2	zero	
21 1201a: 01 47 mv a4, zero		20	12018: 81 40	5		mv	a3, 2	zero	
		21	1201a: 01 47			mv	a4, 2	zero	
		23	1201e: 01 48	3		mv	a6, 2	zero	

文

档

Haawking IDE 中也提供了另外一种方式获得反汇编文件,右键点击项目名称,在弹出的菜单中点击"Properties"按钮,打开项目属性选项卡。"C/C++ Build→Settings→Toolchains",勾选"Create extended listing"。右键项 目名称,在菜单中点击"Rebuild Project"按钮,待项目重新编译完成后,在 项目的 Debug 目录下就能够看到反汇编文件,文件后缀为".1st"。

Properties for hello_haawk	sing	– 🗆 X
type filter text	Settings	↓ ↓ ↓ ↓
> Resource Builders	Configuration: Debug [Active]	V Manage Configurations
Coguing 2 Settings 2 Tool Chain Editor > C/C++ General > MCU Project Natures Project References Run/Debug Settings Task Tags > Validation WikiText	Name: Hawking RISC-V LLVM/GCC Newlib (riscv32-hawking-elf-gcc) Architecture: RISC-V Prefix: riscv32-hawking-elf- Suffix:	v
	Remove command: m Toolchain path: D:\Haawking-IDE-win64-V1.5.0(haawking-tools/compiler/Haawking-RISCV-LLVM/bin (to change it use the global or workspace preferences pages or the groject properties page) Build tools path: (to change it use the global or workspace preferences pages or the project properties page) © Create flash image Ø Create extended listing 4	Restore Defaults Apply
?		Apply and Close Cancel

中

科

旲

芯

1.2.6 如何切换已有工程 FLASH 和 RAM

在新建工程的时候,一般都会选择了 RAM 或者 FLASH 执行。

BR			x c	
New HaawKing P Input project bas	roject ic information. Filter and select corresponding chip dem	10.		
Target: <se< th=""><th>ect or type filter text></th><th>ł</th><th>~</th><th></th></se<>	ect or type filter text>	ł	~	
Connection: Haa	wking HX100V2	~ Ve	erify	
盲 template-inf	D			
Project name:	hello_haawking			
Use default le	ocation:			
Locatio	n: D:/IDE_Workspace_1_4_5/test_rename_02	Bro	wse	
Compiler versio	n: Haawking RISC-V LLVM	~ Mo	ore	
RAM/FLASH:	Program Running in RAM	✓ Bro	wse	
type file text	Program Running in RAM		~	
Empty P	ojects	-		
Basic	Project 创建工程时选择RAM/FLASH工程		\$	

对于已有工程,首先通过右键项目名称->"Properties"按钮,打开项目的 Properties 面板。

在"C/C++ Build"->"Settings"->"Tool Settings"->"Target Processor" -> "Choose FLASH/RAM"选项中,通过下拉框来对工程的 RAM 和 FLASH 属性进行 切换,切换 RAM/FALSH 工程选项之后,点击"Apply and Close"来使配置生效。 切换配置后,重新编译工程即可。

注: 操作步骤可以看下方图片



1.2.7 头文件、函数、变量无法跳转到声明位置

在 Haawking IDE 中,使用右键菜单->"Open Declaration"、快捷键 F3 或按住 Ctrl 键并且单击可以跳转到函数、变量声明的位置。



如果出现无法跳转或者跳转位置出错的问题,请按照下面步骤操作:

步骤一:如果用户使用 F3 快捷键方式,那么先尝试使用右键菜单->"Open Declaration"方式,如果仍然出错,按步骤二操作;如果能够正确跳转,请用 北京中科吴芯科技有限公司 20

户检查系统的F3快捷键是否被其它软件占用。

步骤二:点击菜单栏中的"Project"->"C/C++ Index"->"Rebuild"按钮,待 IDE 右下方的进度条完成后,即可正常使用。



1.2.8 Haawking IDE 不支持在任务栏中启动

Haawking IDE 尚未支持在 Windows 10 任务栏中启动,请用户勿将 Haawking IDE 固定到任务栏。



北京中科昊芯科技有限公司

1.2.9 IDE 路径包含中文字符或者空格

Haawking IDE 不支持路径中包含中文字符或者空格,如果解压路径包含中 文字符或者空格,则将无法启动 Haawking IDE,同时会弹出如下错误信息:

ERROR PATH



1.3 HaawkingIDE 工程目录介绍

Haawking	IDE 的工程目录如下表	::
----------	--------------	----

目录结构	对应下图中得序号	Haawking IDE 工程目录
用户文件目录	1	src 文件夹下
生成文件目录	2	Debug 文件夹下
驱动库	4	common、f2802x_common、 f2802x_header 文件下
调试器配置文件	3	board 下的. cfg 文件
链接文件	5	ldscripts 文件夹下的.ld 文件

注意: Haawking IDE 是把 FLash 和 RAM 的地址空间单独配置的,但不建议用

户对".1d"文件进行修改。

Haawking IDE 的工程目录如下图:



1.4 Map 文件简要说明

1、如何通过 map 文件查看程序分布

打开"map"文件, 找到"Linker script and memory map"这一段, 如下

图: 725 *default* 0x00000000000000 0xfffffffffffffff 726 inker script and memory map 729 730.AdcRegs 731 *(.AdcRegs) 1) 它下面的内容如图所示: 0x0000000007c01ee 0x30 ./src/main.o text main 0x00000000007c01ee main .text.DisableDog 0x00000000007c021e 0x26 ./haawking-drivers/haawking-dsc28027-board/f2802x common/source/f2 0x0000000007c021e DisableDog .text.handle_trap 0x0000000007c0244 0x1e ./haawking-drivers/haawking-dsc28027-board/common/syscalls.o 0x0000000007c0244 handle_trap .text_clean_bss 0x0000000007c0262 0x48 /haawking-drivers/haawking-dsc28027-board/common/syscalls.o clean bss 0x0000000007c0262 text init data 0x0000000007c02aa 0x6c /haawking-drivers/haawking-dsc28027-board/common/syscalls.o 0x0000000007c02aa _init_data _text_init_data_l1 0x0000000007c0316 0x70 /haawking-drivers/haawking-dsc28027-board/common/syscalls.o _init_data_l1 0x00000000007c0316 .text_init_data_m0 0x0000000007c0386 0x68 /haawking-drivers/haawking-dsc28027-board/common/syscalls.o _init_data_m0 0x0000000007c0386 text, init data m1 0x00000000007c03ee 0x70 /haawking-drivers/haawking-dsc28027-board/common/syscalls.o 0x00000000007c03ee _init_data_m1 text. DisableDog 0x00000000007c045e 0x26 /haawking-drivers/haawking-dsc28027-board/common/syscalls.o DisableDog 0x00000000007c045e 2) 其中的格式如表格中显示:

段名称	段运行地址	段大小、段所在文件
	函数运行地址	函数名称
	函数运行地址	函数名称

注意: 当只有段名称, 没有段运行地址的时候, 该段已经被编译器删除了。

1.5 HaawkingIDE 暂未支持的功能

1.5.1 工程重命名

Haawking IDE 暂不支持工程重命名功能。



1.5.2 实时波形显示

Haawking IDE 暂不支持实时波形显示功能。

中 科 旲 芯 档 文

2 Haawking IDE 使用

2.1 打开 Haawking IDE

双击根目录下 Haawking-IDE-win64-V1.5.1. exe 可执行文件,即可打开 Haawking IDE.

								-	• ;	< ?
Haawking-IDE-win64-V1.6.0	~	ō	р į	健索"Haav	wking-	IDE-win	64-V1.6.0)"		7
^ 名称 ^	修改日	日期		类型			大小			
 eclipse haawking-tools Haawking-IDE-win64-V1.6.0 ③ open_ide ③ open_ide_spike 1/1 北京中科異芯科技有限公司-Haawking-II 	2021, 2021, 2021, 2021, 2021, 2021, D 2021,	/12/24 /12/11 /3/13 1 /8/30 1 /8/19 1 /12/27	15:48 17:20 6:25 6:19 6:03 17:55	文件 文件 应用 Wind Wind Wind	夹 程序 dows 扣 dows 扣 PDF 文	处理 处理 档	3,33 8,48	7 KB 2 KB 1 KB 4 KB		
该执行文件会配置好需要	的工具	,如	ı "JDF	ζ"、	"0r	oen00	CD"、	编译	器和	链接
器等。用户可	以 i	先	择	默	认	的	I	作	空)Ē
<u>(C:\Users\username\haawkin</u> 준。	<u>g-work</u>	spac	ce-V1.	<u>. 5)</u> ,	也可	可以枯	艮 据需	害要选	择其	他路
注意事项:工作空间所在	的路径	不允	许包	含中了	文字	符!				-
Haawking IDE V1.6.0 Launcher Select a directory as workspace Haawking IDE V1.6.0 uses the workspace	e directory t	to stor	e its pref	erences	and de	evelopm	nent artifa	acts.	×	
Workspace: C:\Users\mgd\haawking-wo	orkspace-V	/1.6				~	Browse.			
Use this as the default and do not ask	again									
					[Laur	nch	Cano	cel	

在 IDE 启动成功后会弹出下方的欢迎界面。





2.2 新建工程

Haawking IDE 不支持 "&"、" |" 等运算符号以及中文字符进行命名, 推荐使用字母开头,数字+字母+下划线的组合进行命名。

目前 Haawking IDE 支持两种创建工程的方式,下面对两种方式分别进行介绍。

第一种方式: 使用 Haawking Project 创建工程

首先点击"File->New->Projects",打开新建工程的窗口,选择"Haawking Project->Haawking Project"选项,点击"Next"进入下一个界面。

I	科	旲	芯	文
🔢 New F	Project		11_11	
Select a v	vizard			
Create a	Haawking Project			\square
Wizards:				
type filte	r text			
> 🦢 C/ > 🍎 H: > 🍎 Tr > 🍎 E:	/C++ aawking Project Haawking Project acing amples			
Show	All Wizards.			
?	< Bacl	< Next >	Finish	Cancel

档

在新的界面中依次选择芯片型号、选择 RAM/FLASH、选择 "Basic Project"

作为模板、输入项目名称,点击 Finish 按钮即可。

Target:	<select filter="" or="" te<="" th="" type=""><th>xt></th><th>V HXS320F28</th><th>3027</th><th></th><th>~</th></select>	xt>	V HXS320F28	3027		~
Connection:	Haawking HX100V2			~	Verify	
templa	te-info				1 c nym	
- compia		输入项目名	称			
Project na	me: hello_haawking					
Use def	ault location:					
Lo	cation: C:/Users/mgd/h	aawking-worl	cspace-V1.5		Browse	
Compiler v	ersion: Haawking RISC-	v选择RAM/	FLASH	~	More	
RAM/FLAS	H: Program Runnin	g in RAM		~	Browse	
type file to	ext		DSC28027 Basic	Project	6	1
⇒ Em ~ ③ BA:	pty Projects SIC Basic Project 选择Basic Project				v	

点击 Finish 后,等待一会后 IDE 左侧的 "Project Explorer"面板中会出现之 前新建的工程。

注: 当一个新的空间中创建第一个工程时,可能会花费较长的时间,此时可 以查看 IDE 右下角的任务进度提示,如下图 2 所示。

	科	旲	芯	文		档		
Proje	ect Explorer 🛛				7 8			
∨ h	ello_haawking							
> 6	Includes							
> 🖻	haawking-drive	ers/haawking	-dsc28027-bo	ard				
> 2	src 🗧							
>@	haawking-drive	ers						
>0	b lib							
🛐 Problems 🕮	🧟 Tasks 📮 Console 🔲 Prop	perties						7 % □
Description	^	Resource Path	Location	Туре				
							创建工程的任务	进度提示
							Create Project (70%)	-
							Create Project: (79%)	-

第二种方式:使用 C Project 创建工程

点击"File->New->Projects",打开新建工程的窗口,选择"C/C++->C Project"选项,点击"Next"进入下一个界面。

New Project			6_20		×
Select a wizard				and the second second	\diamond
Create a new C	project				4
Wizards:					
type filter text					
> 🗁 General					
✓ 🥭 C/C++	4				
C Proje	Project				
Haawk	ing Project				
> 🕞 Tracing					
> 🧽 Examples					
	-i				
Snow All Wiza	ras.				
0	D 1	N	er. 4 ().	~	1
U	< Back	Next >	Finish	Canc	el

在打开的"C Project"面板中,选择"Haawking RISC-V C Project",输入项目名称"hello_haawking",点击next 按钮。

中 科 昊 芯 文 樘	当
-------------	---

C Project			×	
Project Create C project of selected type			2	
Project name: hello_haawking	输入项目名			
✓ Use default location				
Location: C:\Users\mgd\haawking-work	space-V1.3.0_13\hello_ha	Browse		
Choose file system: default 🗠				
Project type:	Toolchains:			
> 🗁 GNU Autotools Y 🕞 Executable	RISC-V Cross GCC			
Hello World ANSI C Project	选择Haawking RISC-	V C Pro	oject	
 Haawking RISC-V C Project Shared Library Static Library Makefile project 				
☑ Show project types and toolchains on	y if they are supported on t	he platfo	rm	
	1			
? < Back Ne	xt > Finish	Cance		$\lambda!V$

 \mathbf{V}

在新出现的面板中,选择芯片型号、RAM/FALSH工程,点击"Next"按钮

	C Project	6 <u>—</u> 0	
	Project settings Select the Haawking bo	oard and define project options.	Ď
	切换	换 开 发板	
	Board:	Haawking DSC28027 Board	~
	Content:	Empty (add your own content)	~
	RAM/FLASH	Program Running in RAM	~
	Use system calls:	切换RAM/FLASH工程 Freestanding (no POSIX system calls)	~
	Trace output:	None (no trace output)	~
	Check some warnings		
	Check most warnings		
//	Enable -Werror		
	Use -Og on debug		
	Use newlib nano		
	?	< Back Next > Finish	Cancel

在新出现的面板中,点击"Next"按钮。

中	科	旲	芯	文	档		
		Recent					
		C Project					
		Select Config Select platfo	urations rms and configuration	ons you wish to de	ploy on	Ď	
		Project type: Toolchains: Configuratio	Executable RISC-V Cross G ns:	cc			
		☑ 🛞 Debu ☑ 🛞 Relea	ig ase		Select Deselec	t all	
					Advanced s	settings	
		Use "Advanc	ed settings" button	to edit project's p	roperties.		
		Use "Manag	e configurations" bu	ittons either on to	olbar or on pro	operty pag	
			~	占土Nievt控卸			L
		?	< Back	Next > Fi	nish	Cancel	
在新闻	面板中占击	"Finish	" 按钮		ζĸ		
ן זעג בבן		TIIISI	12 山	. X			
	RS C	Project					×
	GNU Sele	RISC-V Cross	Toolchain n and configure	e path			2
	Tool	chain name: 19	RISC-V LLVM	GCC Newlib (r	iscv32-haav	wking-elf-gc	
	Tool	hain path:	:\HaawkingIDE	\Haawking-IDI	E-win6 <mark>4-V1</mark> .	3.0\h Brov	vse
				110			
Ň							
	511						
				1	点击Fins	n创建完成	
	?	< B	ack Ne	ext >	Finish	Cance	1
	ė.			914			

如果出现下图中的弹窗, 点击"Open Perspective"按钮。





创建成功后效果如下图所示

🔁 • 🖬 🕞 📎 •	¶ ▼ 🗟 💣 ▼ 🖄 ▼ 🖻 ▼ 🞯 ▼ 🕸 ▼ 🔘 ▼ 🥵 ▼ 💁 ▼ 🎒 🖉 ▼ 📴 Π	
Project Explorer 🖾	🖻 🔄 🍞 🕴 🗖	
> 📂 hello_haawking		
1		
1	成功创建的工程	
1	成功创建的工程	
1	成功创建的工程	

2.3 编译

2.3.1 整个工程编译

右键单击工程名,选择 Build Project 编译整个工程。如果编译出现错误, 请检查环境变量是否设置正确。

 \wedge



编译成功后,在"hello_haawking->Debug"目录下,会出现可执行文件 hello_haawking.elf。示例程序的路径为:



2.3.2 单个文件编译

中

在所需编译的文件处鼠标右击,然后左键点击"Build Selected File(s)" 即可编译选中文件。

	2802x_a	dc.c		
	2802× cr 280:	New	>	
> 🗔 f	280;	0		
> .c) f2	280;	Open	410.0110.044	
> 🔂 f2	280;	Show In	Alt+Shift+W>	
> 💽 f2	280;	Open With	>	
> 💽 fź	280;	Show in Local Terminal	>	
> 💽 f2	280:	Сору	Ctrl+C	
> 💽 f2	280:	Paste	Ctrl+V	
> 💽 f2	280: 🗙	Delete	Delete	
> 🖻 f2	280;	Source	,	
> <u>S</u> ie	er_s	Move		
> <u>s</u> ie	er_u	Bename	F2	
> <u>.s</u> if	r_se			
> <u>.s</u> if	r_ur 🚵	Import		
→ f2802x	hei 🕍	Export		
> 👝 inclu	Ide	Build Project		
> 🔁 sour	rce	Refresh	F5	
	SFAC	Index		
	ne is			
READN	AE.r	Resource Configurations	,	
src		Build Selected File(s)		
HK_all	incl	Clean Selected File(s)		
🔥 include	he_he	Profiling Tools	>	2
💼 initializ	e-h	Team	,	
interru	pts	Compare With	,	
💼 main.c		Beplace With		
💽 timer0	ba	Run C/C++ Code Analysis		
h timer0	ba: 👋	Validate		
C TM165	0_II 🗠	validate		
h TM165	0_11	Properties	Alt+Enter	
Delaure				

2.4 如何生成库文件

在 Haawking IDE, 工程编译后,默认会在工程的 Debug 目录下生成 elf 格式的文件,该格式的文件可用于调试和烧入,如果用户需要生成静态库文件,可按照下方步骤操作。

2.4.1 前置工作

如果用户使用的是 Haawking IDE1.5.4 版本前创建的工程,首先需要按照下面的操作进行配置。

- 1. 选中项目,右键弹出菜单,点击最下方的"Properties"选项。
- 2. 依次选择 C/C++ Build->Settings->Build Steps 选项。
- 3. 删除 Post-build steps->Command 文本框中的内容。
- 4. 点击 Apply and Close 按钮保存配置。

注: 用户也可以按照下方的图片进行配置。

Properties for GPIO_TES	τ -	
	Settings	$\diamondsuit \bullet \diamondsuit$
Resource Builders C/C++ Build 1	Configuration: Debug [Active]	nfigurations
Build Variables Environment Logging Settings (2) Tool Chain Editor C/C++ General MCU Project Natures Project Natures Project References Run/Debug Settings Task Tags Validation WikiText	Tool Settings S Toolchains Build Steps Build Artifact B Binary Parsers Error Parsers Pre-build steps Command: Description: Post-build steps Command: Introduction Post-build steps Post-buil	~ ~ ~
Wikilext	riscv32-hawking-elf-objdump -D '\$(ProjName).elf' > '\$(ProjName).asm';riscv64-unknown-elf-objdump -D '\$(ProjName).elf' > '\$(ProjName).s'; Description: 删除文本框中的内容	~
	保存配置	

使用 IDE 生成静态库时,工程的 Source Location 中配置的目录中的所有.c 和.S 文件均会被包含在内。

北京中科昊芯科技有限公司

而使用 Haawking IDE 建立的工程,默认会包含下方的两个 Source Location 目录,因此,如果用户生成的静态库不想包含驱动库,那么需要在 Source Location 中删除驱动库目录。

(1) src 目录

(2) Haawking-drivers/haawking-dsc28027-board 目录(驱动库所在目录)

W Debug - Hojeer Explorer M
V 😂 GPIO_TEST
> 🎛 Archives
> 🗊 Includes
> 😕 haawking-drivers/haawking-dsc28027-board
> 🐸 src
> 🔁 Debug
✓
> 📂 haawking-devices
> 😕 haawking-dsc28027-board
> 🕞 ldscripts
🗙 package.json
🗙 xmake.json
> 🗁 lib
GPIO_TEST.launch

下面演示如何删除 Source Location 中的目录:

1. 选中项目,右键菜单,点击最下方的"Properties"按钮,打开项目的 "Properties"面板。

2. 依次选择 C/C++ General->Paths and Symbols->Source Location

3. 在 Source Location 面板中选中待删除的路径,点击右侧的 Delete 按钮,最后点击 "Apply and Close"按钮保存配置即可。

4. 此时,在 Source Location 中删除的目录中的文件就不会被打包进静态库中。
| 中 | 科 | 旲 | 芯 | 文 | 档 |
|---|---|---|---|---|---|
| | | | | | |

Properties for GPIO_TEST		— 🗆 X
type filter text	Paths and Symbols	⇔ • ⇔ • 8
 > Resource Builders > C/C++ Build > C/C++ General > Code Analysis 	Configuration: Debug [Active]	 ✓ Manage Configurations
Documentation File Types	Source folders on build path:	Add Folder
Formatter Indexer	> CONCINENTIAL AND CONTINUES AND A CONTINUES A	Link Folder
Language Mappings Paths and Symbols (2)		Edit Filter
Preprocessor Include Profiling Categories		Delete
 > MCU Project Natures Project References Run/Debug Settings Task Tags > Validation WikiText 		\$
< >>		Restore Defaults Apply
0		6 Apply and Close Cancel

2.4.2 生成静态库

1. 选中项目,右键弹出菜单,点击最下方的"Properties"选项,打开项目的"Properties" 面板。

2. 依次选择 C/C++ Build->Settings->Build Artifact, 打开"Build Artifact" 面板。

3. 在"Build Artifact" 面板中, 点击"Artifact Type"的下拉列表, 选择"Static Library"选项, 最后点击"Apply and Close"保存配置即可。

注: 用户可以按下方图片操作

中	科	旲	芯	文	档

Properties for GPIO_TES	EST	— 🗆 X
type filter text	Settings	<> ▼ <> ▼ §
 > Resource Builders > (C/C++ Build 1) Build Variables Environment Logging > Settings 2) Tool Chain Editor > (C++ General > MCU Project Natures Project Natures > Project Natures > Project Natures > Project Natures > Task Tags > Validation > WikiText 	Configuration: Debug [Active] Tool Settings Toolchains P Build Steps Duild Artifact Binary Parsers Error Parsers Artifact Type: Artifact actension: Static Library Artifact extension: Static Library Output prefix: Iib	Manage Configurations
?	(5) A	oply and Close Cancel

4. 配置完成后,用户重新编译工程,在工程下的"Archives"选项或 Debug 目录下,即可看到.a 结尾的库文件。

注: "Archives"下的.a 文件与 Debug 目录下的.a 文件是同一个, "Archives"下的.a 文件是 Debug 目录下的.a 文件的映射。



2.4.3 使用静态库

用户生成静态库后,如果想要使用该静态库,需要按照下方的步骤进行配置。

1. 将生成的库文件放至工程下的 lib 文件夹中。

👺 Test_Pointer	
> 🎇 Binaries	
> 🗊 Includes	
> 😕 haawking-drivers/haawking-dsc280)27-board
😕 src	
> 🔁 Debug	
> 📂 haawking-drivers	
✓ 🤭 lib	
ibDSC28027_init_call.a	
🗟 libGPIO_TEST.a	生成的静态库
ibhaawking-dsc28027-segger_li	b_imc.a
🗟 liblQmath.a	
Test_Pointer.launch	

2. 打开项目的"Properties"面板, 选择 C/C++ Build->Settings->Tool Settings, 打开"Tool Seeings"面板。

3. 点击 GNU RISC-V Cross C Linker->Libraries, 在右侧第一个面板中点击

🖣 按钮。



4. 在弹窗中输入静态库的名称, 点击 OK 即可。

注:如果要添加名为"libGPIO_TEST.a"的库,那么在弹窗中输入的名称只需要"GPIO_TEST",即忽略 lib 前缀和.a 后缀。

Enter Value	×
Libraries (-l)	
GPIO_TEST	

5. 最后点击"Apply and Close"按钮保存配置即可

2.5 编译优化介绍

2.5.1-01 级别优化

- 1、删除无用参数
- 2、合并冗余指令
- 3、内联函数
- 4、有条件地消除无用库调用
- 5、循环不变量:找出循环不变量并使用"代码移动 (code motion)"将其移出循环体
- 6、删除无用循环
- 7、展开循环
- 8、循环拆分
- 9、循环简化
- 10、删除冗余的指令
- 11、删除无用位
- 12、循环旋转,修改循环结构,使其便于编译器优化

2.5.2 -O2 级别优化

在 O1 的基础上, 增加

- 1、删除未使用的异常处理信息
- 2、尾调用优化
- 3、重新组合表达式
- 4、删除无用的全局变量
- 5、合并重复的全局常量
- 6、剥离未使用的函数原型

2.6 生成 bin 格式文件

在 Haawking IDE 中,工程编译后,在 Debug 目录下默认会生成 hex 格式的 文件,如果用户需要 bin 格式的文件可以按下方步骤操作。

1. 选中项目,右键弹出菜单,点击底部的"Properties"选项,弹出项目的 "Properties 面板"。

GPIO_TE	New	>	24
	Go Into		13
诜中项目, 7	右键弹出苹单		p4
21-247	Open in New Window		55
	Show In	Alt+Shift+W >	160
	Show in Local Terminal	>	1
	Сору	Ctrl+C	10
(D)	Paste	Ctrl+V	6
×	Delete	Delete	E
	Source	>	6
	Move		3
	Rename	F2	54
2-2	Import		55
<u>1</u>	Export		56
	Rebuild Project		17
	Build Project		18
	Clean Project		19
สา	Refresh	F5	1
	Close Project		12
	Close Unrelated Project		13
	Build Targets	>	74
	Index	>	75
	Build Configurations	>	16
	Profiling Tools	>	7
0	Rup Ar	Ś	8
*	Debug Ar	,	19
Sec.	Profile As	>	10
	Team	>	62
	Compare With	>	12
	Restore from Local History		
	Run C/C++ Code Analysis		Cor
	Validate		Bu
	Carfana		CV3
	Comgare	,	text
-	Source	,	285

2. 在弹出的菜单中按下方文字提示或图片操作即可。

(1) 点击左侧菜单中的 C/C++ Build->Settings

(2) 在菜单右侧选择 Tool Settings->GNU RISC-V Cross Create Flash Image->General

(3) 在右侧的面板中,点击 Output file format (-O)选项的下拉菜单,选择 Raw binary 按钮。

(4) 点击下方的"Apply and Close"按钮即可。

Properties for GPIO_TEST		- 🗆 X
type filter text	Settings	🗢 - 🖒 - §
type filter text Nesource Builders < C/C++ Build ① Build Variables Environment Logging Settings C/C+ Ceneral > MCU Project Natures Project Natures Project Natures Nr/Oebug Settings Task Tags > Validation WikiText	Settings Configuration: Debug [Active] Tool Setting: Toolchain: Build Steps Build Artifact in Bina Target Processor Output file format (-O) Inte Warnings Other sections ; J data Warnings Warnings Wiscellaneous Workings Warnings Wiscellaneous Warnings Miscellaneous Workings Miscellaneous W	ゆ ▼ ☆ ▼ ⑧ Manage Configurations ry Parsers ● Error Parsers ● HEX HEX HEX HEX Terole S-record torole S-record torole S-record torole S-record torole S-record torole S-record TERT
	Other flags	
		Restore Defaults Apply
0		Apply and Close保存配置 Apply and Close Cancel

3. 重新编译工程, 在工程下的 Debug 文件夹中即可看到".bin"结尾的文



2.7 下载功能

Haawking IDE 仿真器下载功能暂不支持启动调试后二次下载程序。

2.7.1 前置工作

由于该功能并不兼容 Haawking IDE 1.5.3 版本前创建的工程,因此,如果用 户希望在旧版工程上使用该功能,首先需要按照下面的步骤更新工程中的配置文件。

1. 首先使用 V1.5.3 版本以后的 IDE 创建一个与旧工程芯片型号一致的工程。

🔁 Project Explorer 🛛	□ 🕏 🍸 🕴 🗖	
> 👺 HXV1.5.4_28027_PROJECT		

创建一个与旧工程芯片型号一致的工程

2. 选中该项目,右键弹出菜单,点击"Properties"选项,打开"Properties"

面板。

3. 选择左侧的"Resource"选项,点击右侧的 [☎] 按钮,在文件管理器打开 项目。

Properties for HXV1.5.4	28027_PROJECT	– 🗆 X
type filter text	Resource	
 Resource Builders C/C++ Build C/C++ General MCU Project Natures Project References Run/Debug Settings Task Tags Validation WikiText 	Path: /HXV1.5.4_28027_PROJECT Type: Project Location: D:\IDE_Workspace_1_5_4\test_document_01\HXV1.5.4_28027_1 Last modified: 2021年12月24日下午6:55:34 Text file encoding Inherited from container (GBK) Other: GBK Store the encoding of derived resources separately New text file line delimiter Other: Windows) Other: Windows	PROJECT Show In System 打开该工程

4. 进入项目目录,拷贝项目中的.haawking 文件,将其复制到旧工程中,如

果提示是否覆盖,点击覆盖即可。

中

	2024 (42.02.4.4.2.55		
settings	2021/12/24 18:55	又件夹	
haawking-drivers	2021/12/24 18:55	文件夹	
lib	2021/12/24 18:55	文件夹	
src	2021/12/24 18:55	文件夹	
.cproject	2021/12/24 18:55	CPROJECT 文件	68 KB
desc .	2021/12/24 18:55	DESC 文件	1 KB
haawking	2021/12/24 18:55	HAAWKING 文件	1 KB
.project	2021/12/24 18:55	PROJECT 文件	1 KB
HXV1.24_28027_PROJECT	2021/12/24 18:55	LAUNCH 文件	6 KB

将.haawking文件拷贝到旧版工程中, 如果提示是否覆盖,选择覆盖即可。

2.7.2 使用 IDE 下载程序

2.7.2.1 直接下载工程中的程序

如果用户希望直接下载工程中的程序,那么首先选中待下载的工程,然后直接点击菜单栏上的 · 按钮,即可开启下载流程。

注: 默认下载工程中 Debug 目录下的 elf 文件,如果用户需要下载.hex 格式 文件,请看 2.7.2.2 小节。



点击按钮后, IDE 中会出现一个进度条提示下载进度, 当下载完成后, 进度 条会自动关闭, 如果下载途中没有任何警告, 则代表下载成功。



2.7.2.2 下载指定位置的程序

用户如果希望下载任意位置的程序(.elf 文件或.hex)文件,首先需要一个工程 作为下载时的配置文件所在工程,因此用户可以新建一个空的 FLASH 工程作为 配置文件所在工程。

注: 待下载文件所在位置不允许包含中文字符

首先点击菜单栏上下载按钮的▼选项,然后点击"Select Program to Download"选项。



在出现的弹窗中选择待下载的文件,支持.elf 格式文件和.hex 格式文件的下载,然后选择确定按钮即开启下载流程,下载过程中如果未出现任何警告,则 代表程序下载成功。

如果用户希望查看使用"memory"视图查看芯片内容,可以看文档 3.9 小节,使用"Debug Without Download"功能。

注: 下载文件所在位置不要包含中文字符。

下载面板解释:

Program File: 下载文件所在路径

Browse...: 选择文件资源管理器中的文件

Browse Projects...:选择工作空间中的文件

Configuration: 下载时使用的配置文件所在工程, 配置文件用于解锁芯片、

确定芯片型号等。

BR			>
Program File:	C:\Users\mgd\Desktop\HX_DSC28027_ECAP.elf	Browse	Browse Projects
Configuration:	HXV1.5.4_28027_PROJEC选择待下载的文件	1	
		确定	取消
	F	点击确定开	F始下载

2.7.2.3 关闭下载前编译工程

下载功能默认会勾选 Z Build Project Before Download 选项,即下载前

先编译工程,如果用户不需要编译工程,可以再次点击该按钮,取消下载前编译。



2.7.2.4 配置芯片解锁密码

1. 首先右键工程,选择 Debug as->Debug Configurations,打开工程的调 试面板

2. 选择工程的调试配置,激活"Debugger"选项卡,在"Debugger"选项 卡中配置密码后,点击"Apply"按钮使配置生效,点击"Close"按钮关闭调试 配置面板。

1 🖻 🖗 🗎 🗶 🖻 🏹 🔹	Name: TEST DOWNL	OAD 027 FLASH		
ype filter text	Main 🕸 Debugg	uer Startup & Source Common B SVD Path		
C GDB OpenOCD Debugging	OpenOCD Setup	2		
TEST_DOWNLOAD_027_FLA	Start OpenOCD	locally		
🖡 Launch Group 🕕	Executable path:	\${openocd_path}/\${openocd_executable}	Browse	Variables
	Actual executable:	D:\Haawking-IDE-win64-V1.5.4\eclipse\\\haawking-tools\openocd\	bin/openoco	d.exe
		(to change it use the global or workspace preferences pages or the	project proj	perties page
	GDB port:	3333		-
	Telnet port:	4444		
	Tcl port:	6666		
	Tcl port:	6666		
	Tcl port: Password #1:	6666 FFFFFFF FEFEFEFE		
	Tcl port: Password #1: Password #2:	6666 FFFFFFF FFFFFFF		
	Tcl port: Password #1: Password #2: Password #3:	6666 FFFFFFF FFFFFFFF FFFFFFFF		
	Tcl port: Password #1: Password #2: Password #3: Password #4:	6666 FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF		
	Tcl port: Password #1: Password #2: Password #3: Password #4: Config options:	6666 FFFFFFF FFFFFFF FFFFFFFF FFFFFFFF FFFFF	sc28027.cfg	击Apply
	Tcl port: Password #1: Password #2: Password #3: Password #4: Config options:	6666 FFFFFFF FFFFFFF FFFFFFFF FFFFFFFF -f "haawking-drivers/ aawking-dsc28027-board/board/haawking-d	sc28027.cfg	击Applý
	Tcl port: Password #1: Password #2: Password #3: Password #4: Config options:	6666 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFF	sc28027.cfg	击Apply
	Tcl port: Password #1: Password #2: Password #3: Password #4: Config options:	6666 FFFFFFF FFFFFFF FFFFFFF FFFFFFF -f *haawking-drivers/waawking-dsc28027-board/board/haawking-d -f *haawking-drivers/waawking-dsc28027-board/board/haawking-d	sc28027.cfg	击Apply

2.8 导入工程

在菜单栏点击"File->import"



在"Import"面板中,选择"General->Existing Projects into Workspace" 选项,点击"Next"

Select	`
Create new projects from an archive file or directory.	2
Select an import wizard:	
type filter text	
✓ 🦻 General	
C Archive File	
😢 Existing Projects into Workspace	
🕞 File System	
Preferences	
📮 Projects from Folder or Archive	
> 🔁 C/C++	
> 🔁 Install	
> 🔁 Oomph	
> 🧀 Run/Debug	
> 🤁 Team	

由于待导入的工程有文件夹、压缩包两种格式,因此导入的方式也有所不同,

下面分别就导入文件夹格式工程和压缩包工程进行介绍。

导入文件夹格式的工程时,在 Import 面板中点击"Browse"按钮选择要导入的工程,建议勾选"Copy projects into workspace"按钮,点击"Finsh"按钮即可。

注: 如果不勾选 "Copy projects into workspace" 按钮,请确保工程所 在的位置不包含中文字符。

port Projects ielect a directory to search for existin) Select root directory: G:\DSC28027) Select archive file: 'ojects: [] 015_HX320F28027_HRPWM (G:\)	点击Browse按钮选择得 g Eclipse projects. '例理\/DE_1_3_0\(015_HX3: ~)	Browse	
elect a directory to search for existin) Select root directory: G:\DSC28027) Select archive file: rojects: 015_HX320F28027_HRPWM (G:\	g Eclipse projects. /例理\/DE_ <u>1_3_0\(015_HX32</u> ~)	Browse	
) Select root directory: G:\DSC2802) Select archive file: rojects: 015_HX320F28027_HRPWM (G:\	7例程\IDE_1_3_0\015_HX32 >	Browse Browse	
) Select archive file: rojects: 2 015_HX320F28027_HRPWM (G:\	~	Browse	
rojects:			
015_HX320F28027_HRPWM (G:)			
	,DSC28027例程\IDE_1_3_0\01	Select All	
		Deselect All	
		Refresh	
c	>		
Options			
Search for nested projects			
Close newly conted projects upo	on completion		
Hide projects that already exist in t	the workspace		
Working sets 将上柱守入当制	的工作空间		
Add project to working sets		New	
Working sets:	~	Select	
	点击Finish	按钮	
	Number Children		

导入压缩包格式的工程时,在 Import 面板中勾选 "Select archive file" 按钮, 然后通过 "Browse" 按钮选择待导入的工程,点击 "Finish" 按钮即可。

注: 压缩包格式的工程仅支持.zip 格式。

Import Projects Select a directory to search for existing Eclipse projects.
O Select root directory:
● Select archive file: G:\DSC28027例理\/DE_1_3_0\HX_DSC2 ∨ Browse ProtectAG选Select archive file按钮
点击Browse选择待导入的工程压缩包。 Deselect All Refresh
Options Search for nested projects Copy projects into workspace Close newly imported projects upon completion
Options Search for nested projects Copy projects into workspace Close newly imported projects upon completion Hide projects that already exist in the workspace Working sets Working sets Working sets Select

2.9 导出工程

中

2.9.1 导出压缩包格式工程

1、在菜单栏点击"File -> Export"

<u>F</u> ile	<u>E</u> dit <u>S</u> ource Refactor	<u>N</u> avigate	Se <u>a</u> rch	<u>P</u> rojec
	New		Alt+Shi	ft+N>
	Open File			
	Open Projects from File Sy	/stem		
	Recent Files			>
	Close Editor		Ctr	l+W
	Close All Editors		Ctrl+Shif	t+W
	Save		CI	trl+S
	Save As			
	Save All		Ctrl+Sh	ift+S
	Revert			
	Move			
	Rename			F2
8	Refresh			F5
	Convert Line Delimiters To	,		>
۵	Print		CI	trl+P
	Import	Export		
\langle	Export			
	Properties		Alt+E	inter
	Switch Workspace			>
	Restart			
	Exit			

2、在"Export "面板中,选择"General->Archive File"选项,点击"Next"

🔀 Export — 🗆 🗙
Select
Export resources to an archive file on the local file system.
Select an export wizard:
type filter text
 ◇ Coperative File ◆ File System ○ Preferences > ⊘ C/C++ > ⊘ Install > ⊘ Run/Debug > ⊘ Team > ⊘ Tracing > ⊘ XML
第二步 第二步

3、在新面板中选中待导出的工程,然后点击"Browse"按钮选择要导出的 位置,最后点击"Finish"按钮即可。

Export	
Archive file Export resources to an archive file on the le	ocal file system.
> ☑ C hello_haawking 选择待导出的工程	Ø.cproject Ø.desc Ø.awking Ø.project Ø.awkinglaunch Ø.openotd.exe.stackdump
The second se	
Filter Types Select All Deselect To archive file: D:\hello_haawking.zip D:\hello_haawking.zip	ct All Browse
Filter Types Select All Deselec To archive file: D:\hello_haawking.zip Options	et All
Filter Types Select All Deselection To archive file: D:\hello_haawking.zip Options Save in zip format	et All Browse 选择导出的路径 ⑧ Create directory structure for files
Filter Types Select All Deselection To archive file: D\hello_haawking.zip Options • Save in zip format • Save in tar format • Compress the contents of the file • Resolve and export linked resources	et All Browse 选择导出的路径 ④ Create directory structure for files ○ Create only selected directories
Filter Types Select All Deselection To archive file: Dt/hello_haawking.zip Options	et All Browse 选择导出的路径 ③ Create directory structure for files ③ Create only selected directories 点击Finish按钮

2.9.2 导出文件夹格式工程

中

在菜单栏点击"File -> Export"

File	Edit Source Refactor	Navigate	Se <u>a</u> rch	<u>P</u> roject		
	New		Alt+Sh	ift+N >		
	Open File					
	Open Projects from File S	System				
	Recent Files			>		
	Close Editor		C	trl+W		
	Close All Editors		Ctrl+Sh	ift+W		
	Save		(Ctrl+S		
	Save As					
5	Save All		Ctrl+S	hift+S		
	Revert					
	Move					
	Rename			F2		
5	Refresh			F5		
	Convert Line Delimiters T	o		>		
<u>e</u>	Print		0	Ctrl+P		
1	Import	Export			1	
-	Export					
	Properties		Alt+	Enter		
	Switch Workspace			>		
	Restart					
	Exit					

2、在"Export"面板中,选择"General->File System"选项,点击"Next"

Export Select Export resources to the local file system.		
Select an export wizard: type filter text General General File System File System File System Frie Sys		
? < Back Next > Fin	ish	Cancel

3. 在新面板中选中待导出的工程,然后点击"Browse"按钮选择要导出的 位置,最后点击"Finish"按钮即可。

中

中 科 吴 心 乂	伯
-----------	---

ect
ect
ect
king ct haawking.launch
ocd.exe.stackdump
Browse
选择工程导出的路径
点击Finish按钮
Finish Cancel

2.10使用 Rebuild Project 功能

Rebuild Project 功能是 Clean Project 功能与 Build Project 功能的封装,使用 Rebuild Project 时,IDE 会先清除工程 Debug 目录下上次编译生成的相关文件, 然后执行全量编译,重新生成可执行文件、反汇编文件等。

1. 选中一个工程,右键弹出菜单,点击 Rebuild Project 选项即可。

> 😂 TEST_DOV	WNIC	04D 027 FLASH		2	* Copyr
		New	>	3	* All rig
		Go Into		4	*
		Open in New Window		5	* Redist
		Show In	Alt+Shift+W >	6	* modif
		Show in Local Terminal	>	7	* met: r
	100	Carry	Chille	8	* notice
·		Сору	cur+c	9	* redistr
	10	Paste	Ctrl+V	10	* notice
	×	Delete	Delete	11	* docun
		Source	>	12	* neithe
		Move		13	* contri
		Rename	F2	14	* this sc
	Res I	Import		15	*
	2	Export		16	* THIS S
		DI TID I .	1	17	* "AS IS
		Rebuild Project		18	* LIMITE
7		Build Project		19	* A PAR
		Clean Project		20	* OWNE
	8	Refresh	F5	21	* SPECI.
		Close Project		22	* LIMITE
		Close Unrelated Project		23	* DATA
		Build Targets	>	24	* THEO

2.11 Debug 目录下的文件介绍

在 Haawking IDE 中,工程编译完毕后,默认会在 Debug 目录下生成多个类型的文件,下面进行介绍。





.elf 文件: elf 后缀的文件程序的调试和烧入。

.asm 文件: asm 后缀的文件是工程的反汇编文件,其中包含吴芯自定义的汇编指令。

.hex 文件: hex 后缀的文件是指 INTEL HEX 格式文件,用于程序的烧入。

.map 文件:map 后缀的文件是编译器经过编译后,生成的程序中信息的映射文件, 该文件中包含程序的内存布局、符号的地址等信息。

.s文件:s后缀的文件同样是工程的反汇编文件,但无法显示昊芯自定义的汇编指令。

2.12 用 Spike 模拟器仿真程序

对于没有开发板的客户,可以使用 Haawking Spike Simulator 来进行简单 程序的仿真,目前只限于 CPU 程序,不包括外设。

Spike 工程的新建、编译与现有芯片一样,这里就不再进行过多介绍。在调试的时候,当前版本的 IDE (V1.5.1版本),需要在每次调试之前,手动执行 一个 bat 脚本,启动 Spike 模拟器;同时建议选择 RAM 工程(其实两者没啥区 别Q)。

Spike 模拟器的启动脚本在 Haawking IDE 的根目录下,双击即可,启动之后效果如下所示。



然后,回到 Haawking IDE 中,参考第三章的调试过程,就可以进行仿真和 调试了。

Þ	科	旲	芯	文	档
			_		

1435 1436 float SIN[360]={}; 1437 void FPU_sin_isa(); 1438⊖ int main(void) 439 { float out;
FPU_sin_isa(); 1440 1441 1442 1443 int i;
for(i=0; i<360; i++)</pre> 1444 1445 //printf("%d %f(%f)\n",i,SIN[i],sinf(Rad[i])); 1446 1447 } 1448 449 return 0; 1449 1450 } 1451 1452 // 1453 1454 🖳 Console 🛛 👭 Registers 🖹 Problems 🜔 Executables 🙀 Debugger Console 📋 Memory test-debug-oocd [GDB OpenOCD Debugging] test-debug-ocd [GDB OpenOCD Debugging] Info : only one transport option; autoselect 'jtag' Info : Initializing remote_bitbang driver Info : connecting to localhost:9824 Info : remote_bitbang driver initialized Info : This adapter doesn't support configurable speed Error: Trying to use configured scan chain anyway... Warn : Bypassing JTAG setup events due to errors Started by GNU MCU Eclipse Error: Trying to use configured scan chain anyway... Warn : Bypassing JTAG setup events due to errors ハハア 1435 1435 float SIN[360]={}; 1437 void FPU sin isa(); 1438 int main(void) 1439 { 1440 float out; 1441 FPU sin isa 1442 1443 int i; 1444 for(ieg) i43 1444 for(ieg) i43 1445 {//print/ float out;
FPU_sin_isa(); int i;
for(i=0; i<360; i++)</pre> //printf("%d %f(%f)\n",i,SIN[i],sinf(Rad[i])); 1446 } 1447 1448 449 return 0: 1450 } 1451 1451 1452 1453 1454 11 -----📮 Console 🛛 👭 Registers 🖹 Problems 🜔 Executables 🖳 Debugger Console 📋 Memory Lest-debug-oocd [GDB OpenOCD Debugging] Info : only one transport option; autoselect 'jtag' Info : Initializing remote bitbang driver Info : Connecting to localhostri9824 Info : remote_bitbang driver initialized Info : This adapter doesn't support configurable speed Error: Trying to use configured scan chain anyway... Warn : Bypassing JTAG setup events due to errors Started by GNU MCU Eclipse Error: Trying to use configured scan chain anyway... Warn : Bypassing JTAG setup events due to errors лялалалалалалалалалаласотрителялалалалалалалалалалалалалала managementations and a set of the set o 46_sin: 47 a 48 49 50 51 52 53 54 55 56 57 58 59 60 -isa-----# -# 61 62 63 64 65 66 67 fsw ft9, 0(t3) addi t3, t3, 4 #blcne t4, -84 addi t5, t5, 1 bne t4, t5, _sin 72 73 74 75 76 77 78 79 flw ft10, 4(sp) flw ft9, 8(sp) flw ft8, 12(sp) flw ft7. 16(sp)

3 调试

3.1 启动调试

当前版本 Haawking IDE 在调试工程的时候,借助于"OpenOCD"和"HX-link"。 在进行调试之前,还需要选择工程对应的调试配置。右键单击工程名,选择 "Debug As→Debug Configurations",或者点击菜单栏上的爬虫图标的下拉按 钮,在下拉菜单中选择"Debug Configurations..."按钮。



在"Debug Configuration"面板中,点击"GDB OpenOCD Debugging"菜单 左侧的展开按钮,选择与工程名称相对应的调试配置,然后点击右下角的"Debug" 按钮,启动调试。

中	科	旲	芯	文	档
· ·			-		

Debug Configurations			- 🗆 X
eate, manage, and run configura	tions		Ť.
i 🖻 🐅 🗈 🗙 📄 🍸 👻	Name: hello_haawking]
/pe filter text	📄 Main 🕸 Debugger 🕨 Startup 🔖	Source 🔲 Common 🔚 SVD Path	
GDB OpenOCD Debugging	Project:		
C hello_haawking	hello_haawking		Browse
	C/C++ Application:		
	Debug/hello_haawking.elf		
1		Variables Search Project	Browse
选择与工程名对应的调试配置近	选项 Build (if required) before launching —		
	Build Configuration: Select Automatica	ally	~
	O Enable auto build	○ Disable auto build	
	Use workspace settings	Configure Workspace Settings	
		古丰Debug开应调试	Apply
ter matched 3 of 3 items		WIII Depugy HAE Higher	19990
2)		Dahua	Class
Ð		Debug	Close

如出现下图所示的界面时,单击"Switch",选择允许调试透视图的切换。

C main.c	X	-	
21 * 2 22 * 1 23 * 1 24 * 2 25 * (26 * 0 27 * 28 * 2 29 * 1 30 * 1 31 **	SPECIAL, EXEMPLARY, LIMITED TO, PROCURE DATA, OR PROFITS; OI THEORY OF LIABILITY, (INCLUDING NEGLIGEI OF THIS SOFTWARE, E Authors: Junning.Wu Email : junning.wu@m FILE : main.c	OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT MENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, R BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT NCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE VEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. hail.haawking.com	^
32			
33 #ir 34 #ir 35 #ir 36 37 38 39⊕ int	nclude "dsc_config.h" nclude <syscalls.h> nclude "IQmathLib.h' t main(void)</syscalls.h>	 Online to spectre suitch This kind of launch is configured to open the Debug perspective when it suspends. This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints. Do you want to switch to this perspective now? 	
40 { 41 42	DisableDog(); //Disal	Remember my decision	
43	wniie(i){	Switch	
45	t L		
46			~
<		*	

当"Console"视图中显示出 OpenOCD 相关信息(如下图 1)时并且跳转到仿真 界面(如下图 2)

3.2 调试模式工具栏介绍

🗂 • 🗟 🖏 🐘 • O • 🏊 • 🥭 🖋 • 🌛 🖻 🔌	■ ■ ■ # 3. ◎ (() ● ≂ () ◎ = () ● () = () = () ● () ● = () ● ()	
🂠 Debug 🛙 🏠 Project Explorer 👘 🐩 🖬		😑 🗇 Varia 🕴 🗣 Brea 🐄 Expr 🛋 Me
C = The for Analysis (Life Uperiod Undersgogin)	25 (INCLUDING NEGLOCENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE 26 (INCLUDING NEGLOCENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE 27 * Authors: Junning Way 28 * Authors: Junning Way 29 * FILE: main.c 30 * FILE: main.c 31 * include "dsc config.h" 32 * include "dsc config.h" 33 * include "dsc config.h" 34 * include "dsc config.h" 35 * include "dsc config.h" 36 * include "dsc config.h" 37 * include "dsc config.h" 38 * include "dsc config.h" 39 * include "dsc config.h" 30 * include "dsc config.h" 31 * include "dsc config.h" 32 * include "dsc config.h" 33 * include "dsc config.h" 34 * include "dsc config.h" 35 * include "dsc config.h" 36 * include "dsc config.h" 37 * include "dsc config.h" 38 * include "dsc	Name Type Image: State of the sta
	49	, `
	Help Lawring (DB OperOCD Debugging) Starting Open On-Chip Debugger(OpenOCD Version V0.19) Now Mass Erasing Flash Mass Erase finished Flash Mass Erase Complete	
■ Resume 按 ⁴	钮的作用为全速运行程序。	
III Suspend 按	钮的作用为暂停程序运行。	
Terminate 3	安钮的作用为终止调试。	
-≫- Step Info 拸	冠 铂作用为单步调试,当遇到函数调用F	时,会进入函数体内
部执行。		
Step Over	安钮的作用也是单步调试,但如果遇到函	函数调用,Step Over
按钮不会进行函数	内部, 而是把函数调用作为一步来执行。	
- Step Return	、)按钮的作用是,当使用 Step Info 按钮注	进入函数调用的内部

执行时,执行完该函数,并且返回上一层函数。

Restart 按钮作用是在调试时复位芯片,使用该功能时有些注意事项,建 议用户使用前先阅读 2.5.3 小节的内容。

➡ Instruction Stepping Mode 按钮: 该按钮用于打开汇编单步调试模式。

3.3 断点的使用及断点视图介绍

3.3.1 如何添加断点

用户启动调试后,在行号前双击即可添加断点,如下图所示。



3.3.2 如何删除断点

将鼠标移动到断点上,再次双击即可删除该断点。

3.3.3 使用 Breakpoints 视图管理断点

在 Haawking IDE 中, "Breakpoints" 视图负责管理所有项目的断点。



3.3.4 Breakpoints 视图功能按钮介绍

X 按钮: 该按钮用户删除选中的断点。

💥 按钮: 删除 Breakpoints 视图中的所有断点。

🚰 按钮: 跳转到断点所在的文件。

▲ 按钮:点击该按钮后,所有的断点都会被跳过,即程序并不会在断点处停止。

☑ 复选框: 断点前的 ☑ 复选框用于设置断点的生效/失效状态,当取消选中复选框时,断点处于失效状态。

	main.c [line: 62]
0.2	main.c [line: 63]
2 🔎	main.c [line: 65]
	main.c [line: 70]

3.3.5 FLASH 工程使用断点的注意事项

因为在 Flash 工程中,只允许用户最多加入两个断点(硬件断点),但是在进入 Debug 模式后,默认会在进入"main()"函数的第一个函数的位置加入断点,因此在进入 Debug 模式之前,工程中最多存在一个生效的断点。

断点的位置见下图:



因此,在FLASH项目启动调试前,用户需要在Breakpoints视图中管理工程已有的断点,下面提供几种方式。

1. 在 Breakpoints 视图中删除多余的断点或删除全部断点。

2. 在 Breakpoints 视图中点击 2、按钮, 使多余的断点处于失效状态。

注意:如果在 Debug 模式下,再加入第三个断点,或进入 Debug 模式之前就 断点数量就已经超过了三个,那么控制台就会报如下错误:



导致芯片无法再继续运行,需要点击关闭按键 B 关掉进程,再移除所有断点后 再重新进入 Debug 模式。

3.4 如何使用 Restart 功能复位芯片

在调试过程中,用户可通过菜单栏上的 Sestart 按钮来复位芯片,使 PC 寄存器回到程序的入口位置,但如果用户正在调试的工程是一个 FLASH 工程,那么有一个注意事项,下面进行介绍。

3.4.1 FLASH 工程使用 Restart 功能

FLASH 工程目前最多同时支持两个断点,但是在使用 26 前,项目中最多只 ^{北京中科吴芯科技有限公司} 62 允许一个断点生效,这是因为在点击 ² 后,Haawking IDE 会向 main 函数的第 一行插入一个断点,如果此时项目中已经存在两个生效断点,那么断点数量就超 过了限制,将无法正常使用调试,只能够重新启动调试。

因此,用户在 FLASH 工程中使用 🕹 前,需要在 Breakpoints 视图中,将工程中断点置为失效状态,即点击断点前的复选框。



下面演示如果如果工程中的断点数量大于等于两个时,使用 Restart 功能会 出现的问题,下图的示例工程中,在 46、47 行存在两个断点。



如果此时点击 参 按钮,在 IDE 下方的"Console"面板中,会出现下方的提示,此时调试将无法正常使用,用户需要关闭并重新开启调试。

北京中科昊芯科技有限公司

Error: Couldn't find an available hardware trigger. Error: can't add breakpoint: resource not available



3.5 如何查看变量值

在 Debug 模式下,如果需要查看变量的值可以在右侧窗口拦中点击 "Variables"或"Expressions"窗口标签后查看,窗口标签图片见下:

1				1			-	#C 0
	(x)= Variables 🛛	⁰ Breakpoints	🕵 Expressions	🔤 Disassembly	S Live View	🛃 Peripherals		
100				XX		H.	⇒tş	

这两个窗口都有显示变量值的功能,区别在于"Variables"中的变量是因程序 中声明的变量自动填充的,而"Expressions"中需要用户手动添加需要查看的 变量名,变量名的类型和变量的值会自动显示。

注意:当芯片在全速运行时,这两个窗口中的变量值不会实时更新,只有当芯片处于暂停的状态或单步调试的模式时,可查看变量值。如果需要查看变量的 实时状态值,用户可参阅"2.5.7使用实时刷新功能"章节。

			🕵 Expressions 🛛		
(x)= Variables 🖾			Expression	Туре	Value
Name	Туре	Value	(×)= i	volatile unsigned int	2048
		2040	(×)= temp	int	2048
(×)= temp	Int	2048	👍 status		
(×)= i	volatile unsigne	2048			
(×)= status	Uint16	7514		24.5500 (2555220) (25502)	
				、用户可手动汤	动变量名

"Variables" 和 "Expressions" 两个窗口的内容显示如下图:

如果进入 Debug 模式中,没有显示这两个窗口时,用户可在工具栏的"Window" -> "Show View"中查找。位置如下图



中

或在工具栏的"Window"->"Show View"->"Other..."中的"Debug"下单击 选中"Variables"或"Expressions",再点击"Open"就可以显示在 IDE 的右 侧窗口栏。操作图片如下:

Ν	lindo	W He					Show View		\times
	N E A	lew Window ditor > ppearance >	q trl.o	• • 🥭 🔗 • 🦽 { = ⊡ f2802x_exampl - 2	la ▼	•	type filter text		
	S	how View 🔷 🚿	00	Breakpoints	Alt+Shift+Q, B		V 🔁 Debug		^
	Pe	erspective >		Console	Alt+Shift+Q, C		Breakpoints		
	N	l <mark>avi</mark> gation >	夺	Debug		14	The Debug		-
	P	references		Debugger Console			Debugger Console		
-	56		G .	Debug Sources			🖽 Debug Sources		
	57	/*关闭着门狗	-	Disassembly	Alter Chiffer O. J.		Disassembly		
>	58	DisableDog(10	Error Log	Alt+Shift+Q, L	ЬU	Executables		
	59		₩.	Expressions		1	ଙ୍କୁ Expressions		
1	60	/*时钟初始化	n	Memory			Memory		
	61	InitSysCtrl();	0	Memory Browser			Memory Browser		
	62		-	Modules			Modules		
	63	/*解锁*/	8	Outline	Alt+Shift+Q, O		CS Resources		
	64 CF	CsmUnlock()	品	Peripherals			🛃 Peripherals		
	65	/*CDI037744(Problems	Alt+Shift+Q, X		1919 Registers		
	57	GPIO INITO	-0	Progress			a Signals		
	68		B	Project Explorer			Strace Control		
	69	/*给数组变量	1010	Registers		Ьŭ	(x)= Variables		
	70	for(i = 0; i <		Signals			A Docker		¥
	71	Buffer[i		Templates					
	72		5	Terminal					
	73	/*调用FlashW	20	Trace Control	3	100			
	74	status = Flas	(x)=	Variables	Alt+Shift+Q, V	The last	Open	Cancel	
		<		Other	Alt+Shift+Q, Q				

3.6 如何查看外设寄存器的值

在 Debug 模式下,如果需要查看芯片外设寄存器的值可以在右侧窗口拦中点 北京中科吴芯科技有限公司 65

击"Peripherals"窗口标签后查看,窗口标签图片见下:

						-
(×)= Variables ⊠	e Breakpoints	Service Expressions	🔛 Disassembly	S Live View	Peripherals	
					£.	⇒t; [=

打开后会显示该芯片的所有外设模块名称。点击对应模块名称前面的方块,会在 IDE下面 "Memory"的窗口队列中显示该名称。操作方式如下图:

	discourse of the second		Console dia Registers	Progress Problem	s 😈 executables 🙀	Debugger console
Peripheral	Address	Description	Monitors 👍 💥 🎉	B GPIO: 0xD800 23	New Renderings	
ADCREGS	0x00001400	Analog to Digital Convetor Peripheral	A 0.7-1050			
ADCRESULT	0x0000149C	Analog to Digital Convetor Peripheral	UX/C1000	Register	Address	Value
CAP	0x0000D000	Serial Peripheral Interface (SPI) Peripheral	GPIO	✓ 品 GPIO	0x0000D800	
CMP-A	0x0000C000	Comparator		> IN GPAMUX1	0x0000D800	0x00000000
🗆 🚼 DMA	0x00001C00	Direct Memory Access Peripheral		> IIII GPAMUX2	0x0000D804	0x00000030
TRASH	0x007AF800	Flash Register Peripheral		> IIII GPBMUX1	0x0000D808	0x00000000
GPIO	0x0000D800	General Purpose Input/Output Controller (GPIO) Peripheral		> IIII AIOMUX1	0x0000D80C	0x0000000
□ 器 12C	0x0000E400	Inter-Integrated Circuit Module		> 100 GDACTRI	0-00000910	0~0000000
E RE	0x00001800	Peripheral Interrupt Expansion		MIN CONCTRU	0.00000014	0.0000000
C R QEP	0x0000D400	Quadrature Encoder Pulse		S HE GPBCIRE	0x0000D814	0x0000000
CI 🚼 SCI	0x0000E000	Serial Communications Interface		> GPAQSEL1	0x0000D818	0x0000000
D 🛃 SPI	0x0000E800	Serial Peripheral Interface (SPI) Peripheral		> IIII GPAQSEL2	0x0000D81C	0x0000000
SysCtrl	0x0000DC00	Serial Peripheral Interface (SPI) Peripheral		> IIII GPBQSEL1	0x0000D820	0x0000000
B RePWM1	0x0000B000	Enhanceed Pulse-Width Modulation (PWM) Peripheral		> IIII GPADIR	0x0000D824	0x00000000
🗌 🚼 ePWM2	0x0000B400	Enhanceed Pulse-Width Modulation (PWM) Peripheral		> IIII GPBDIR	0x0000D828	0x00000001
EPWM3	0x0000B800	Enhanceed Pulse-Width Modulation (PWM) Peripheral		> IIII AIODIR	0x0000D82C	0x0000000
🗌 🚼 ePWM4	0x0000BC00	Enhanceed Pulse-Width Modulation (PWM) Peripheral		> IN GPAPUD	0x0000D830	0x00000FFF

GPIO 模块中的全部寄存器的寄存器名称、地址和数值会显示在该窗口中。 接下来具体说明一下"SysCtr1"和"GPIO"模块中寄存器状态的查看过程。 当程序编译完成后,进入 Debug 模式后,可单步调试 → → 或打断点在 "InitSysCtr1();"函数之后,当运行完"InitSysCtr1();"函数后,可查看倍 频系数和分频系数是否已经写入"PLLCR"和"PLLSTS"寄存器中,数据显示见 下图:

Console 🔠 Registers	🗝 Progress 📳 Problems 🜔 Exec	utables 📓 Debugger Cons	sole 🚺 Memory 🛛	Console 🐰 Registers	🖏 Progress 🖹 Problems 🜔 Ex	ecutables 🖳 Debugger Con	sole 🚺 Memory 🛛
Monitors 🝦 💥 💥	🚼 SysCtrl: 0xDC00 🛛 💠 New R	enderings		Monitors 👍 🐭 🎉	SveCtrl: 0xDC00 8	Renderings	
0x7c1060	Register	Address	Value	0. 00 T	as system oxpecto to price	Renderingsin	1.55.4
GPIO	✓	0x0000DC00		0x/c1060	Register	Address	Value
SysCtrl	> Mil XCLK	0x0000DC00	0x00000002	GPIO	> IIII INTOSCITRIM	0x0000DC1C	0x00002800
	✓ MIN PLLSTS	0x0000DC04	0x00000181	SysCtrl	> INTOSC2TRIM	0x0000DC20	0x00002800
	PLLLOCKS	[0]	0x1	2	> 1818 CLKCTL	0x0000DC24	0x00006400
	PLLOFF	[2]	0x0		V IN PLICE	0x0000DC28	0x000000A
	MCLKSTS	[3]	0x0		DIVE	[3:0]	0xA
	MCLKCLR	[4]	0x0			[010]	0.0
	OSCOFF	[5]	0x0		DIVIN	[4]	0x0
	. MCLKOFF	[6]	0x0		DIVM	[6:5]	0x0
	OIVSEL	[8:7]	0x3		> III PLLLOCKPRD	0x0000DC2C	0x0000FFFF
	IN NURMEDTE	[15]	UXU		> IN LPMCRO	0x0000DC30	0x000000FC

查看数据后,继续运行 GPIO 配置程序 "GpioDataRegs.GPATOGGLE.bit.GPIOO = 1;"和 "GpioDataRegs.GPATOGGLE.bit.GPIOO = 0;"后,数据显示见下图:

E Console 🐰 Registers	🖷 Progress 🔝 Problems	s 🚺 Executables	🔀 Debugger Console	🚺 Memory 🛛	Console 🚻 Registers	🖷 Progress 🖹 Problem	s 🚺 Executables	R Debugger Console	🚺 Memory 🛛
Monitors 🝦 💥 💥	🔒 GPIO: 0xD800 🛛 📢	New Renderings.	.)		Monitors 💠 💥 💥	🔚 GPIO: 0xD800 🛛 🤘	New Rendering	s	
♦ 0x7c1060	Register	Address	Value		♦ 0x7c1060	Register	Address	Value	
GPIO	V 1918 GPADAT	0x0000D838	0x300F10FF		GPIO	Y 1010 GPADAT	0x0000D838	0x300F10FE	
SysCtrl	GPIO0	[0]	0x1		 Sysctri 	GPIO0	[0]	0x0	
	GPIO1	[1]	0x1			. GPIO2	[1]	0x1	
	🖁 GPIO2	[2]	0x1			GPIO3	[3]	0x1	
	B GPIO3	[3]	0x1			🔐 GPIO4	[4]	0x1	
	CPIO4	[4]	0x1			GPIO5	[5]	0x1	

注意:当芯片在全速运行时,这两个窗口中的变量值不会实时更新,只有当芯片处于暂停的状态或单步调试的模式时,可查看变量值。如果需要查看变量的 实时状态值,用户可参阅"2.5.8使用实时刷新功能"章节。

北京中科吴芯科技有限公司

3.7 使用 Memort 视图查看存储器中的值

3.7.1 如何查看指定地址处的数据

在 Debug 模式下,可输入存储器或寄存器的地址查看该地址的数值。具体操 作见下图:

■ Console 1 Mer Monitors 4	nory 원 1889 R	egisters 🖬 🕅	Monitor Memory nter address or exp nonitor: Dx7df000 (OK	× eression to 人 Cancel	俞入地址,并 击"OK" ■
完成这两步操作	作就可以查利	 	的数据了。		
Console 🚺 Memory	🛛 🚻 Registers	Rogress	🖞 Problems 🚺 E	xecutables 🛛 🖳 D	eb <mark>ugg</mark> er Console
Monitors 🖕 💥 💥	0x7df000 : 0x7Df	F000 <hex> 🔀</hex>	🗣 New Rende	rings	
0x7df000	Address	0 - 3	4 - 7	8 - B	C - F
	007DF000	FFFFFFF	FFFFFFF	000A0000	010A0000
	007DF010	020A0000	030A0000	040A0000	050A0000
	007DF020	060A0000	070A0000	080A0000	090A0000
	007DF030	0A0A0000	0B0A0000	0C0A0000	0D0A0000
	007DF040	0E0A0000	0F0A0000	100A0000	110A0000
	007DF050	120A0000	130A0000	140A0000	150A0000
	007DF060	160A0000	170A0000	180A0000	190A0000
	007DF070	1A0A0000	1B0A0000	1C0A0000	1D0A0000
	007DF080	1E0A0000	1F0A0000	200A0000	210A0000
	007DF090	220A0000	230A0000	240A0000	250A0000

注意:当芯片在全速运行时,Memory视图中的值不会实时更新,只有当芯片 处于暂停的状态或单步调试的模式时,可查看变量值。

3.7.2 如何导出 Memory 视图中的数据

1. 点击 Memory 视图中的["]按钮, 弹出"Export Memory"弹窗。

中	科	旲	芯	文	档
1		~	0	~	

🔄 Console 🕮 Registers 🖷 Progress 🖹 Problems 🜔 Executables	Wemory & go Der	bugger console	A 1		S
Monitors 🚽	🗱 🙀 0x7c0000 : 0x7	C0000 <hex> 🕅</hex>	🔶 New Renderi	ngs	
♦ 0x7c0000	Address	0 - 3	4 - 7	8 - B	C - F
	007C0000	81400141	81410142	81420143 该按钮	81430144
	007C0010	81440145	81450146	81460147	81470148
	007C0020	81480149	8149014A	814A014B	814B014C
	00700030	73400230	814D014E	63C30200	01400702
	007C0040	00009382	E2037390	52309701	85EE9381
	007C0060	617A1702	84FF1302	D27D1372	02FC7325
	007C0070	40F18545	6370B500	13161501	32921701
	007C0080	84FF1301	21786111	6F000065	6D7106C2
(1) Format: 导出的文件格式 (2) Start address: 起始地址				Ĺ	
 (1) Format: 导出的文件格式 (2) Start address: 起始地址 (3) End address: 结束地址 (4) Length: 自定义导出数据的 (5) File Name: 导出文件的名称 	り长度 称				
 (1) Format: 导出的文件格式 (2) Start address: 起始地址 (3) End address: 结束地址 (4) Length: 自定义导出数据的 (5) File Name: 导出文件的名 	竹长度 称				
 (1) Format: 导出的文件格式 (2) Start address: 起始地址 (3) End address: 结束地址 (4) Length: 自定义导出数据的 (5) File Name: 导出文件的名 	勺长度 称 出文件格王	t			>
 (1) Format: 导出的文件格式 (2) Start address: 起始地址 (3) End address: 结束地址 (4) Length: 自定义导出数据的 (5) File Name: 导出文件的名 Export Memory Format: RAW Binary ✓ 输 Start address: 0x7c0000 End a	勺长度 称 出文件格式	t 4000	Length	: 16384	
 (1) Format: 导出的文件格式 (2) Start address: 起始地址 (3) End address: 结束地址 (4) Length: 自定义导出数据的 (5) File Name: 导出文件的名 Export Memory Format: RAW Binary ✓ Start address: 0x7c0000 End a File name: C:\Users\mgd\Desktop\Memory	勺长度 称 出文件格元 ddress: 0x7c4	t 4000	Length Brows	: 16384 .e	

3.8 如何进入汇编调试

1、在工具栏点击^{i→}"Instruction Stepping Mode"按钮。

 Eile
 Edit
 Source
 Refactor
 Navigate
 Search
 Project
 Run
 Download
 Window
 Help

 \square \square </td

2、查看右侧"Diassembly"显示栏

44	科	关	Ϋ́	X	,	 ゴ				
(x)= Variables	Breakpoints 🙀 E	xpressions 📑 Mo	dules 🔛 Disass	embly 🛛	S Live View	Reripherals			-	
				E	inter location	n here 🗸 🗸	1 2 6	\$ Q	11	000
 000128d2: 60 000128d4: 000128d6: 000128d8: \$62 000128da: 000128da: 000128de: 63 000128e2: 000128e2: 000128e3: 000128e3: 000128e3: 000128e3: 	jal 0x1253e InitPieCtrl(); jal 0x1248a lui a0,0x10 addi a0,a0, IER_DISABLI sw a0,-28 jal ra,0x121 IFR_DISABLE lw a1,-28(sw a0,-32 mv a0,a1 ial ra_0x121	<pre>e <initsysctrl> a <initpiectrl> -1 E(0xffff); (s0) 80 <ier_unset> E(0xffff); s0) (s0) 88 <ifr_unset></ifr_unset></ier_unset></initpiectrl></initsysctrl></pre>								
3、单步汇	二编调试继续	卖使用工具 	栏中的	"Step ▶	Into"	按钮 	1			

3.9 使用 Debug Without Download 功能

功能介绍: 该功能用于不擦除芯片的内容时启动调试,调试启动后,用户可以使用 Memory 视图查看芯片中的内容,或者调试芯片中的程序。

3.9.1 前置工作

由于该功能并不兼容 Haawking IDE 1.5.3 版本前创建的工程,因此,如果用 户希望在旧版工程上使用该功能,首先需要按照下面的步骤更新工程中的配置文件。

1. 首先使用 V1.5.3 版本及以后的 IDE 创建一个与旧工程芯片型号一致的工



2. 选中该项目,右键弹出菜单,点击"Properties"选项,打开"Properties" 面板。

3. 选择左侧的"Resource"选项,点击右侧的 按钮,在文件管理器打开项目。

				0
type filter text	Resource Path: /HXV1.5.4_28027_PROJECT Type: Project Location: D:\IDE_Workspace_1_5_4\test_document_01\HXV1.5.4_28027_PROJECT Last modified: 2021年12月24日 下午6:55:34 Text file encoding Inherited from container (GBK) Other: GBK Store the encoding of derived resources separately New text file line delimiter Inherited from container (Windows) Other: Windows	♀ I I I I I I I I I I I I I	• c>	▼ 8 System
0		(

4. 进入项目目录,拷贝项目中的.haawking 文件,将其复制到旧工程中,如

果提示是否覆盖,	点击覆盖即可。

A .		SIA TER	ALC: N
石 林	修改口期	类型	大小
settings	2021/12/24 18:55	文件夹	
haawking-drivers	2021/12/24 18:55	文件夹	
lib	2021/12/24 18:55	文件夹	
src	2021/12/24 18:55	文件夹	
🔐 .cproject	2021/12/24 18:55	CPROJECT 文件	68 KB
🖹 .desc	2021/12/24 18:55	DESC 文件	1 KB
🗈 .haawking	2021/12/24 18:55	HAAWKING 文件	1 KB
🚽 .project	2021/12/24 18:55	PROJECT 文件	1 KB
HXV1. 4228027_PROJECT	2021/12/24 18:55	LAUNCH 文件	6 KB
将.haawking文 如里提示是否覆	件拷贝到旧版工程中,		

3.9.2 启动 Debug Without Download

1. 右键工程,选择"Debug As->Debug Configurations",打开工程的 Debug

配置面板。

Ł	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5	17 AS IS AND ANT EAPRESS OR IMPL 18 * LIMITED TO, THE IMPLIED WARRAN 19 * A PARTICULAR PURPOSE ARE DISCL 20 * OWNER OR CONTRIBUTORS BE LIAI 21 * SPECIAL, EXEMPLARY, OR CONSEQU 22 * LIMITED TO, PROCUREMENT OF SU
0	Close Unrelated Project Build Targets Index Build Configurations Profiling Tools	> > > >	23 * DATA, OR PROFITS; OR BUSINESS IN 24 * THEORY OF LIABILITY, WHETHER IN 25 * (INCLUDING NEGLIGENCE OR OTHE 26 * OF THIS SOFTWARE, EVEN IF ADVIS 27 *
\$	Debug As Profile As Team Compare With Restore from Local History	> > > >	Debug Configurations pail.haawking.c 30 * FILE : main.c 31 ************************************
*	Run C/C++ Code Analysis Validate		0 items Description

2. 选中工程的调试配置(与工程同名),激活右侧的"Startup"选项卡, 勾选"Debug Without Download"复选框即可。

由于 Debug Without Download 功能需要启动调试,因此需要一个调试信息 所在的文件,默认情况下会使用当前工程下的 elf 文件作为调试信息所在文件, 用户也可以指定任意的文件作为调试文件。

(1) Use project binary 选项: 使用工程下的 elf 文件作为调试文件

(2) Use file 选项: 用户自定义一个 elf 文件作为调试文件

Debug Configurations	- 0
reate, manage, and run configurations	
type filter text 📄 Main 🏇 Debugger 🕨 Startup 🦃	Source 🔲 Common 😹 SVD Path
▼ ⓒ GDB OpenOCD Debugging ⓒ TEST_DOWNLOAD_027_FLASH □ Initial Reset. Type: init	tartup选项卡
a Launch Gr改先中项目调试配置 monitor dsc28027 probe 0 monitor reset init monitor dsc28027 unlock 0	
Enable ARM semihosting	
Target Configuration	
Debug Without Download	
Use project binary: TEST_DOWNLO	4D_027_FLASH.elf
O Use file:	Workspace File System
Run/Restart Commands	
3勾选Debug Without Download Pre-run/Restart reset Type:	halt (always executed at Restart)
	~
	~
Set program counter at (hex):	
	Develop Areals
ilter matched 3 of 3 items	Revert Apply
<i>O</i>	Debug Close

3. 点击 Debug 进入调试。

3.9.3 Debug Without Download 功能注意事项

如果用户希望使用"Debug Without Download"功能调试芯片中的程序, 那么用于启动"Debug Without Download"的工程中的程序需要与芯片中的程 序保持一致,否则可能无法达到正常调试的效果。

3.10使用实时刷新功能

由于 Haawking IDE 的 Expressions 视图不支持在芯片运行时查看变量的值,因此,用户想在芯片运行时查看变量的值需要使用"Live View"视图。

3.10.1 实时刷新视图使用注意事项

实时刷新视图仅支持添加全局变量,并且仅支持添加全局变量名称。
 例:

如果程序中定义了一个名为"test_struct_val"的变量,那么对于实时刷新视 图 来 说 , "test_struct_val" 是 一 个 合 法 的 变 量 名 , 但 是 "test_struct_val.test_int_member"则是一个非法的变量名。

1. struct {
 2. int test_int_member;
3. }test_struct_val;

2. 请勿向实时刷新视图添加占用内存过大的变量,否则可能会导致 IDE 出现卡顿。

3. 实时刷新视图默认处于暂停刷新状态,如果用户需要使用实时刷新,则
 需要手动点击 by 按钮来开启实时刷新。

4. 实时刷新模块目前仅支持变量的查看,不支持变量的写入。

5. 实时刷新模块目前不支持变量的波形图显示。

6. 实时刷新视图目前仍处于开发阶段,部分功能存在问题,具体的问题请
中 科 昊 芯 文 档

看 3.9.8 小节中的内容。

3.10.2实时刷新视图按钮介绍

👺 Start Refresh/Stop Refresh 按钮用于开启、暂停实时刷新

Add Expression To LiveView 按钮用于向实时刷新视图添加变量,它会 弹出一个输入框,让用户输入待添加的变量名称

✗Remove Selection Variable 按钮用于删除选中变量

🖗 Remove All Variables 按钮用于删除实时刷新视图中所有的变量

⁶⁹ Change Refresh Interval 按钮用于调整实时刷新的速度,默认的刷新间

隔为 200ms

xpression 🖕 Add new variable	^{Type} 暂停、开	Yalue 启实时刷新	Advess
1	弹	出输入框用于添加变量	*///
点击输入变量名	;称	删除选中的	变量
		家时刷新视图中所有	1变量
		调整实时	刷新的速度

3.10.3 向实时刷新视图添加变量

向实时刷新视图添加调试变量的方式有两种。

第一种方式:未开启 Debug 时,通过点击实时刷新视图中的 👻 按钮,或者 直接编辑 "Add new variable"元素来输入变量名称。

中	科	昊 7	芯文	1 档		
(x)= Variables	💩 Breakpo	. 🙀 Expressi	Disasse	S Live View 🛛	🔓 Peripher	 % (0)
Expression	w variable	Туре		Value		Addre
_/	通过编辑A	dd new varia	ble元素输入变	量名		

第二种方式为,在开启 Debug 后,在编辑视图中选中要添加的变量,然后右击,选择"Add To LiveView"按钮,同样会弹出输入框,框中已经包含刚才选中的变量名称,点击 OK 即可将其加入实时刷新视图进行调试。

SU A FILE Main C	,	In _ILOULA_USUCIUS-S	M ILOULA_DOC	~		
31 ************	****	*****	******	*****/		
32						
33 #include "dsc of	con	fia.h"				
34 #include <sysc< td=""><td>alle</td><td>h.</td><td></td><td></td><td></td><td></td></sysc<>	alle	h.				
35 #include "IQma	Ø	Undo	Ctrl+Z			
36		Revert File				
37 int test_int_val;		Save	Ctrl+S			
38		Open Declaration	F3			
39 ^e struct {		Open Type Hierarchy	F4			
40 int test_int_m	1	Open Call Hierarchy	Ctrl+Alt+H			
41 }test_struct_val;		Quick Outline	Ctrl+O			
42		Quick Type Hierarchy	Ctrl+T			
		Explore Macro Expansion	Ctrl+#			
45 InitSveCtrl()		Toggle Source/Header	Ctrl+Tab	a known state		
46		Open With	>	G KHOWH State.	-	
47 while(1){		Show In	Alt+Shift+W >			
48 test int va	of	Cut	Ctrl+X			
49 }		Сору	Ctrl+C			
50	13	Paste	Ctrl+V			
51 return 0;		Quick Fix	Ctrl+1			
52 }		Source	Alt+Shift+S >			
53		Surround With	Alt+Shift+Z >			
54 //		Refactor	>			
55		Declarations		-		
		References	ĺ.			
Console 🛛 👭 Registe		Search Text	Ś	ebugger Console 🚺 Memory 🛷 Searc	ch	
hello_haawking [GDB Open		Search rexe				
Starting Open On-Ch		Build Selected File(s)				
Now Mass Erasing		Clean Selected File(s)		_		
Flash Mass Erase finis		Resource Configurations	>			
Flash Mass Erase Cor	6	Add To LiveView				
	õ	Run As	× ,			
	松	Debug As				
		Profile As	>			
int main/void	N					1
int main(void)	/					
InitSysCtrl();	11	Initializes the Sv	stem Con	trol registers to a know	vn state	
1	1.00		X+Y Ad	d Variable To LiveView	×	
while(1){			The Va	ariable Name		
test int v	al+	+:				
1			test_i	nt_val		
}						
return 0				OK	Cancel	
ictuin 0,						
ł			L			
				点击OK按钮,即可将变	2 量添加至实时刷新视图	<u>[]</u>
2.20						

中 科 吴 芯 文 档

3.10.4 删除实时刷新视图中的变量

如果想要删除一个变量,首先在实时刷新视图中选中该变量,然后点击视图中的 送按钮即可删除选中的变量。

💴 Variab 🤏 Break	🙀 Expre 🛋 Mod	ul 🔛 Disass 🕃 Live	V 🛛	Perip	
				🦻 🕂 🗙	* 🙀 🙆
Expression	Туре	Value	Remo	ove Selector	Variable
(×)= test_int_val	int	0			0x1260
🐈 Add new variable					

3.10.5切换变量输出格式

实时刷新视图支持多种输出格式, 默认的输出格式为按照其类型进行解析的数据。如果想要切换变量的输出格式,首先需要选中一个元素,然后右击,在 弹出的菜单中选择"Change Show Format"即可切换显示格式。

Expression	Туре	选中变量并且右键弹	伸出菜单
(x)= test_int_val Add new variable	int_	Add To LiveView Remove	
		Change Show Format	Default
	1		Select Q-Value Q-Value(24) Q-Value(30) BINARY OCTAL DECIMAL HEX

"Change Show Format"菜单中选项的含义:

Default	该按钮是将变量按照其类型进行解析
Select Q-Value	该按钮将会弹出一个输入框,在其中输入具体的 IQ 数值

中 科 关 心 乂	档
-----------	---

Q-Value(24)	该按钮将会将数据按照 IQ24 格式显示
Q-Value(30)	该按钮将会将数据按照 IQ30 格式显示
BINARY	该按钮将会显示变量数据对应的2进制形式
OCTAL	该按钮将会显示变量数据对应的8进制形式
DECIMAL	该按钮将变量转换为有符号整数进行显示
HEX	该按钮将会显示变量数据对应的 16 进制形式

3.10.6切换刷新间隔

点击视图右上角的 7 按钮,将会弹出输入框用来输入新的刷新间隔,单位 为毫秒,默认刷新间隔为 200ms,实时刷新视图允许的最小间隔为 100ms,用户 通常使用默认的刷新间隔即可。

^{(x)=} Variab ⁹ 9 Break ⁶	🖉 Expre 🛛 🛋 Mod	ul 🔛 Disas	s 🕄	Live V 🛛	<mark></mark> Perip	
					🤒 🕂 🗙	* 🖗
Expression	Туре		Value	Current Re	fresh Interval	[200 ms]
(x)= test_int_val	int		0		/	0x1260
💠 Add new variable					/	
		0	刀换实	时刷新视图	的刷新间隔	i
	•					
Ochange LiveView Ti	me Interval	×				
The Time Interval(100n	ns~100000ms)					
500						
	ОК	Cancel				

3.10.7数据存在变化时的显示效果

如果一个数据距离上次刷新时值有变化,那么该变量的背景色将变为黄色。 北京中科吴芯科技有限公司 76 中 科 昊 芯 文 档

注: 实时刷新视图第一次启动时,默认处于暂停刷新状态,需要手动点击 逐 按钮来开启刷新。

^{(x)=} Variab ⁹ Break	. 🙀 Expre 🛋 N	Modul 🔛 Disass	🕄 Live V 🛛	뭄 Perip	
				🛞 🕂 🗙	* 🙆
Expression	Туре	Va	lue		Addre:
(x)= test_int_val	int	23	971001		0x1260
💠 Add new variable	,				

3.10.8 导出实时刷新视图中的数据

该功能用于导出实时刷新视图中的数据,导出数据的格式为 csv 文件,如下 图所示。

注: 如果要导出实时的数据, 请确保实时刷新视图不处于暂停状态

	А	В	С	D	E
1	arr-[0]	arr-[1]	arr-[2]	arr-[3]	arr-[4]
2	3132912	3132912	3132912	3132912	3132911
3	3132912	3132912	3132912	3132912	3132911
4	3132912	3132912	3132912	3132912	3132911
5	3132912	3132912	3132912	3132912	3132911
6	3132912	3132912	3132912	3132912	3132911
7	3132912	3132912	3132912	3132912	3132911
8	3132912	3132912	3132912	3132912	3132911
9	3132912	3132912	3132912	3132912	3132911
10	3132912	3132912	3132912	3132912	3132911
11	3132912	3132912	3132912	3132912	3132911
12	3132912	3132912	3132912	3132912	3132911
13	3132912	3132912	3132912	3132912	3132911
14	3132912	3132912	3132912	3132912	3132911
15	3132912	3132912	3132912	3132912	3132911
16	3132912	3132912	3132912	3132912	3132911
17	3132912	3132912	3132912	3132912	3132911
18	3132912	3132912	3132912	3132912	3132911
19	3132912	3132912	3132912	3132912	3132911

1. 首先在实时刷新视图中选中待导出的元素,可以按住 Ctrl 键选中多个元

素,	然后右键菜单,	点击	"Start Export	Variable	Value"	按钮。



中 科 昊 芯 文 档

2. 当用户需要暂停导出时,在实时刷新视图右键,点击"Stop Export Variable Value"按钮。

n Type Value int [10] 0x10080 <arr> 0] int Add To LiveView 1] int Remove 3] int Stop Export Variable Value</arr>		Export Count: 50 🍪 🕂 🕷	*	
int [10] 0x10080 <arr> 0] int 1] int Add To LiveView 2] int Remove 3] tint Stop Export Variable Value</arr>	on Type	Value	Value	
0] int Add To LiveView 1] int Remove 3] int Stop Export Variable Value	int [10]	0x10080 <arr></arr>	F.	
1] int Add To LiveView 2] int Remove 3] int Stop Export Variable Value	[0] int	27012250		
2] int Remove 31 int Stop Export Variable Value	[1] int	Add To LiveView		
31 int Stop Export Variable Value	[2] int	Remove		
-1	[3] int	Stop Export Variable Value		
4] in Change Show Format	[4] ip	Change Show Format		
5] int under standard and standard a	[5] int			
6] 🚺 int 0	[6] int	0		
7] int 0	[7] int	0		
8] int 0	[8] int	0		
9] int 0	[9] int	0		

3. 点击"Stop Export Variable Value"按钮后,会弹出保存文件弹窗,保存导出的文件即可。

					▼ 0 ~ Æ	
目织 ▼ 新建文件夹						
🔷 OneDrive - Pers	名称	修改日期	类型	大小		
MIDC MIA	aaa	2021/10/9 16:54	文件夹			
N WPSMM	IISExpress	2021/9/13 10:18	文件夹			
_ 此电脑	Inno Setup Examples Output	2021/12/18 20:06	文件夹			
1 3D 对象	KingsoftData	2021/8/4 14:56	文件夹			
🖪 视频	My Web Sites	2021/9/13 10:18	文件夹			
	PowerShell	2021/10/19 13:39	文件夹			
	Source Insight 4.0	2021/12/14 15:14	文件夹			
	Tencent Files	2021/8/23 9:54	文件夹			
➡ ト载	📙 Visual Studio 2019	2021/9/13 11:18	文件夹			
♪ 音乐	📙 WeChat Files	2021/12/29 9:47	文件夹			
重 桌面		2021/9/13 10:44	文件夹			
🏪 本地磁盘 (C:)		2021/12/28 13:29	文件夹			
🕳 新加卷 (D:)						
🕳 新加卷 (F:)						
🕳 新加卷 (G:)						
U 盘 (H:)						
文件名(N): 2021	年-12月-29日-10-32-28					
保存送刑(T)· csv F	iles (*.csv)					

3.10.9 实时刷新视图存在的问题

编辑变量功能: 如果用户在调试过程中,编辑了"Live View"中的元素,相 当于删除了旧的元素,会出现删除功能存在的问题。同时还可能造成用户卡顿。

中	科	旲	芯	文	档

(x)= Varia 💁 Brea	🛠 Expr 🛋 Modu 🕃	Live 🛛 🔀 Perip 🖓 🗖
Expression	Туре	Value
(x)= <mark>x</mark> =(x)	int	0
🐈 Add new variabl	e	
不建议用户在说	周试过程中,编辑Live	e View中的元素

3.11 如何加密解密

3.11.1 加密操作

我们的 IDE 没有直接设置加密密码的位置,而是通过在一个密码文件中修改 对应的密钥进行加密操作。

如 28027 就是在 f2802x_common/source/f2802x_csmpasswords.c 文件中。 首先在工程目录中,

点击 "haawking-drivers"

-> "haawking-dsc28027-board"

 \rightarrow "f2802x_common"

-> "source"中,找到 "f2802x_csmpasswords.c"文件,双击该文件。 该文件的位置如下:



我们对密钥进行设置后,通过 Debug 的 🐐 🔻 方式对芯片进行加密。

3.11.2 解密操作

完成加密操作后,当需要再次 Debug 的时候,需要进行解密操作。

点击"Debug"的下拉选项

- -> "Debug Configurations" Debug Configurations...
- ->"Debugger"窗口下 隊 Debugger
- -> "Password #1" Password #1: FFFFFFF

中	科		昊 芯	文
\rightarrow	"Password	#2 "	Password #2:	FFFFFFF
\rightarrow	"Password	#3"	Password #3:	FFFFFFF
$- \rangle$	"Password	# 4"	Password #4:	FFFFFFF

输入框中,输入修改后的密钥,修改后见下图。

🖸 🖻 🕫 🐹 🗏 🖻 🏹 👻	Name: hello_haawking-debug-oocd		
type filter text	Mair 🕸 Debugger 🕨 Startup 🦅 Source 🔲 Common 🚼 SVD Path		
 ✓ GDB OpenOCD Debugging 1 € hello_haawking-debug-oocd 	OpenOCD Setup 2		· · · · · · · · · · · · · · · · · · ·
Launch Group	Executable path: \${openocd_path}/\${openocd_executable}	Browse	Variables
	Actual executable: C:\Software\Haawking-IDE-win64-V1.4.0\eclipse\\\haawking-tools\openocd\bin	n/openocd.exe	
	(to change it use the <u>global</u> or <u>workspace</u> preferences pages or the <u>project</u> pro	perties page)	
	GDB port: 3333		
	Tel port: 4444		
	Password #1: 1234FFFF		
	Password #2: FFFFFFF 3		
	Password #3: FFFFFFF		
	Password #4: FFFFDECB		
	Config options: -f *haawking-drivers/haawking-dsc28027-board/board/haawking-dsc28027.cfg	•	^
	Allocate console for OpenOCD Allocate console for the telnet co	onnection	4 .
Filter matched 3 of 3 items		Revert	Apply
?	5	Debug	Close
-> 重占击 "Ann1v"	Revert Apply		
, талла търга			
-> 点击"Debug",	Debug Close		

档

完成解密步骤后,就可以对芯片再次进行调试了。

注意,如果不进行解密操作,就无法对芯片进行调试。当点击"Debug"后, 会有弹窗提示,提示用户芯片是处于加密状态,需要解密后才可以进行调试,弹 出提示如下图。

ERROR	<i></i>
THE DSC28027 WAS LOCK	ED! PLEASE UNLOCK FIRST!

在解密状态下,仍然可以继续配置新的密钥,但一定要记住密钥。如果需要 把芯片的加密状态改为解密状态,需要再将密钥配置为全"F",通过 Debug 相 关的配置就可以实现了。

中	科	旲	芯	文	档
1 2 3 4 5 6 7 8 9 10 11 12 13 14	<pre>#include "F2802x_Device.h" // password file volatile struct CSM_PWL CODI { 0x1234FFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFF, 0xFFFFFFFFFFF, 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</pre>	7 <u>Headerfile</u> Include	File *******/ ") CsmPwl =	1 28 4 5 6 7 8 9 10 11 12 13 13 14	<pre>#include "F2802x_Device.h" // Headerfile Include File /************************************</pre>

中 科	旲	芯	文	档
-----	---	---	---	---

4 审核

版本编号	变更状态	简要说明	变更人	变更日期	批准人	批准日期
						$\langle \rangle$