

**User Guide** 用户指南

目录

# 第1章 认识 TKScope 仿真器

1.1	TKScope 仿真 ARM 种类	5
1.2	TKScope 仿真 ARM 性能	6
1.3	同类比较	7

# 第2章 使用 TKScope 仿真器

2.1	安装驱动	10
2.2	硬件连接	11

# 第3章 Keil RealView MDK 环境仿真方法

3.1	仿真环	际境设置	1	5
3.2	仿真器	<b>縁参数</b> 设置	10	6
	3.2.1	硬件选择	10	6
	3.2.2	主要设置	1	7
	3.2.3	TAP 设置	18	8
	3.2.4	程序烧写	19	9
	3.2.5	初始化宏	22	2
	3.2.6	硬件自检	24	4
3.3	仿真调	間试	24	4
	3.3.1	开始仿真调试	24	4
	3.3.2	仿真调试工具	25	5
	3.3.3	仿真调试结果	20	6

# 第4章 ADS 环境仿真方法

4.1	添加驱动文件		28	
4.2	仿真调	間试	29	
	4.2.1	仿真调试工具	29	
	4.2.2	仿真调试结果	30	

# 第5章 IAR 环境仿真方法

5.1	添加驱动文件		3	2
5.2	仿真调	词试	3	4
	5.2.1	开始仿真调试	3	4
	5.2.2	仿真调试工具	3	5
	5.2.3	仿真调试结果	3	6

# 第6章 TKStudio 环境仿真方法

6.1	仿真环境设置		38	8
6.2	仿真调	]试	39	9
	6.2.1	开始仿真调试	39	9
	6.2.2	仿真调试工具	40	0
	6.2.3	仿真调试结果	41	1

# 第7章 TKScope 仿真器技术支持

7.1	升级方法	43
7.2	联系我们	43
7.3	感谢语	44

# 附录 TKScope 仿真器支持芯片列表

# 第 1 章 认识 TKScope 仿真器



1.1	TKScope 仿真 ARM 种类	5
1.2	TKScope 仿真 ARM 性能	6
13	同米比较	7

# 1.1 TKScope 仿真 ARM 种类

TKScope 嵌入式智能仿真开发平台是广州致远电子有限公司 2008 年隆重推出上市的一款高性能通用型综合仿真开发平台,支持仿 真全系列的 ARM、DSP、AVR、8051、C166、C251、C8051F、MX 等内核;与当前全部主流 IDE 环境无缝嵌接,如 TKStudio、Keil、 ADS、IAR、CCS、RealView、AVRStudio 等,保证您的开发平台始终如一,并具备其高级调试功能。同时,TKScope 内嵌 64 路专业 的逻辑分析仪,zlgLogic 高级软件全面支持。



TKScope 仿真器能够支持 ARM 内核仿真的具体型号有如下几种:

- K系列:K8/K9;
- DK 系列: DK9 / DK10;
- AK 系列: AK100。

本文主要讲解 TKScope 仿真 ARM 内核的性能以及在各个主流 IDE 环境下的使用方法。



目前,TKScope 仿真器支持的 ARM 内核种类如下:

- ARM7: ARM7TDMI、ARM7TDMI-S、ARM7EJ-S、ARM720T;
- **ARM9:** ARM9TDMI、ARM920T、ARM922T、ARM926EJ-S、ARM946E-S、ARM966E-S;
- **ARM11**: ARM1136、ARM1156、ARM1176;
- Cortex- M0/M1/M3: 支持串行调试(SWD)模式;
- XSCALE: PXA255、PXA270。

目前,TKScope 仿真 ARM 内核支持的 IDE 环境如下:

- TKStudio, 致远公司, 中/英文界面, 多内核编译/调试环境, 强大内置编辑器;
- Keil, Keil 公司,英文界面,8051/251/C166/ARM 编译/调试环境;
- ADS, ARM 公司, 英文界面, 全 ARM 内核编译/调试环境;
- IAR, IAR 公司,英文界面,多内核编译/调试环境;
- RealView, ARM 公司,英文界面,全 ARM 内核编译/调试环境。



**TKStudio** 

Keil

ADS

IAR

**RealView** 

# 1.2 TKScope 仿真 ARM 性能

目前,TKScope 支持 ARM7/ARM9/Cortex-M0/Cortex-M1/Cortex-M3/XSCLAE/ARM11 等内核的全系列仿真,包括 Thumb 模式。随 之软件不断升级,后续会支持 Cortex-R4/Cortex-A8 等内核的全系列仿真。

# TKScope 仿真 ARM 内核硬件指标

- USB2.0(High Speed) 高速通讯接口,下载编程 Flash 速度达到 1000KB/S;
- 标准 20-pin JTAG 接口与目标板连接,支持热插拔;
- 检测所有 JTAG 信号和目标板电压;
- 自适应目标板电压,支持宽电压范围 1.8V~5V;
- JTAG 最大时钟 25MHz,可达到极限的调试速度;
- 自动速度识别功能;
- 支持实时 RTCK 同步时钟(自适应时钟);
- 带有硬件自检功能,方便检测排除硬件故障。

# TKScope 仿真 ARM 内核功能特性

- 支持全系列 ARM 内核仿真, ARM7/ARM9/Cortex-M0/Cortex-M1/Cortex-M3/XSCLAE/ARM11 等,包括 Thumb 模式;
- 支持 Cortex-M0/Cortex-M1/Cortex-M3 内核串行调试(SWD)模式;
- 无缝嵌接多种主流 IDE 环境, TKStudio/Keil/ADS/IAR/RealView/SDT 等;
- 支持片内 Flash 在线编程/调试,提供每种芯片对应的 Flash 编程算法文件;
- 支持片外 Flash 在线编程/调试,提供数百种常用的 Flash 器件编程算法文件;
- 支持 NOR/NAND/SPI 等多种接口类型的外部 Flash 编程/调试;
- 支持用户自行添加 Flash 编程算法文件;
- 具备单独烧写 Flash 的独立软件,提高生产效率;
- 支持无限制的 RAM 断点调试;
- 支持无限制的 Flash 断点调试,突破硬件断点数量的限制;
- 采用同步 Flash 技术,快速刷新 Flash 断点,速度如同 RAM 调试一样快捷;
- 支持动态断点,可在运行中任意设置/取消断点;
- 同时支持程序断点和数据断点,便于用户准确跟踪复杂程序的运行;
- 快速单步程序运行,最大 150 步/秒;
- 保证最快最稳定的调试主频变化的目标系统;
- 内置特殊调试算法,可靠调试处于非法状态的 ARM 内核;
- 支持菊花链连接的多器件仿真;
- 基于芯片的设计理念,为数百种芯片提供完善的初始化文件;
- 内置全面的初始化文件解释执行器,可在复位前后/运行前后/Flash下载前后进行灵活的系统设置,包括寄存器设置/ARM 初始化/时钟 设置/延时/信息提示等操作。

# 1.3 同类比较

目前,ARM 仿真器比较流行的是 Ulink2/J-link,但是,TKScope/AK100 仿真器的出现将扭转乾坤,引领 ARM 仿真器的新潮流。 TKScope/AK100 仿真器可以在 TKStudio/Keil/ADS/IAR/RealView 等多种环境下使用,Flash/RAM 断点无限制,支持内/外部 Flash 烧写, 用户可以自行添加 Flash 算法文件等等,这些优质的性能使其完全可以与 Ulink2/J-link 相媲美,甚至还要胜之一筹。

仿真器型号	TKScope K9	AK100	Ulink2	J-link
支持 IDE 环境 ★	TKStudio, Keil, ADS, IAR, RealView	TKStudio, Keil, ADS, IAR, RealView	Keil	Keil, ADS, IAR
RAM 断点	无限制	无限制	无限制	无限制
Flash 断点 <del>*</del>	无限制	无限制	2max	无限制
运行中设置断点	支持	支持	支持	支持
RTCK 同步时钟	支持	支持	支持	支持
烧写内部 Flash	支持	支持	支持	支持
烧写外部 Flash★	支持	支持	支持	支持
支持外部 Flash 种类 <del>*</del>	NOR, NAND, SPI	NOR, NAND, SPI	NOR, NAND, SPI	NOR
用户添加 Flash 算法 <del>*</del>	支持	支持	支持	不支持
RAM 下载速度 ★	650 KB/S	600 KB/S	28KB/S	600 KB/S
JTAG 时钟 <del>×</del>	<=25MHz	<=25MHz	<=10MHz	<=12MHz
快速单步(步/秒)★	≈150	≈150	≈50	≈100
生产厂商	致远电子	致远电子	Keil 公司	Segger 公司

# TKScope K9/AK100/Ulink2/J-link 仿真 ARM 性能对比表

注: ① Ulink2/J-link 仿真器性能参数指标引用其官方网站发布数据,如其更新以网站为准。

② 带★标识为 TKScope/AK100 突出性能。



TKScope K9/AK100/Ulink2/J-link 下载速度实测结果!

TKScope K9/AK100/Ulink2/J-link 实测 NXP LPC2138 芯片,系统环境完全相同,程序代码 512KB,工作时钟为 77MHz。先后使用 这 4 种仿真器分别把程序下载到 Flash 和 RAM 中,记录消耗时间,精确到 0.1 秒。每种下载模式中,分别采用同步时钟和固定时钟 JTAG 1MHz/2MHz/3MHz 测量,记录 4 组时间数据。

# TKScope K9/AK100/Ulink2/J-link 下载速度测试结果(LPC2138)

1				
512KB Flash	JTAG 1MHz	JTAG 2MHz	JTAG 3MHz	RTCK 同步时钟
TKScope K9(秒)	15	9	7.5	6.5
AK100(秒)	15	9	7.5	6.5
Ulink2(秒)	28	28	28	28
J-link(秒)	26	20	下载失败	27
512KB RAM	JTAG 1MHz	JTAG 2MHz	JTAG 3MHz	RTCK 同步时钟
TKScope K9(秒)	6.5	3.5	3	3
AK100(秒)	8	4.5	3.5	3
Ulink2(秒)	23	23	23	23
J-link(秒)	7	4	下载失败	6

注: ① 测试的系统环境完全相同,数据真实记录。

② 编程 LPC2138 芯片 512KB Flash,系统时钟 77MHz。



第2章 使用 TKScope 仿真器

2.1	安装驱动	10
2.2	硬件连接	11

# 2.1 安装驱动

TKScope 仿真器使用之前必须安装驱动程序,否则,无法正常工作!TKScope 仿真 ARM 芯片时,必须安装支持 ARM 仿真的驱动 程序。



用户如果使用 Keil RealView MDK 开发环境,必须把驱动安装到 Keil RealView MDK 目录下;使用其它开发环境, 驱动安装路径可任意选择。

双击 TKScopeSetup\_ARM.EXE,系统会弹出如 图 2.1 所示的对话框,按照提示进行安装即可。

Setup IKScope ARM V3.30	X	
Welcome to TKScope Release 2/2010	<b>TKScope</b> <sup>®</sup>	
This SETUP program installs: <b>TKScope ARM V3.30</b> This SETUP program may be used to update a pr January and a backing a backing application	svious product installation.	
nowever, you should make a backup copy periore proceeding. It is recommended that you exit all Windows programs before continuing with SETUP. Follow the instructions to complete the product installation.		
— TKScope Setup	<< Back Next >> Cancel	

图 2.1 安装驱动提示框

TKScope 仿真器驱动安装完毕,建议用户安装来自微 软的 vc8 实时运行库。双击 vcredist\_x86\_cn.exe,系 统会弹出如图 2.3 所示的对话框。点击【是】,系统会 自动完成安装。

Licrosoft Visual C++ 2005 可再发行组 🔳 🗖 🗙				
请仔细阅读以下许可协议。按 PAGE DOWN 键可查阅协议的余下部 分。				
MICROSOFT软件许可条款				
MICROSOFT VISUAL C++ 2005 RUNTIME LIBRARIES				
本许可条款是 Microsoft Corporation(或您所在地的 Microsoft Corporation 关联公司)与您之间达成的协议。请阅 该本条款的内容。本条款也适用于止述,其中包括您用来接收该软 件的媒体(若有)。本条款也适用于Microsoft: * 更新. * 基于 Internet 的服务和 * 支持服务				
为此软件提供的(除非下述内容附带有其他条款)。如果确实附 🚽				
是否接受本许可协议所列的全部条款?如果选择"否",安装程序 将自动关闭。要安装此软件,必须接受本协议。				
是①				

图 2.3 安装实时运行库

本文示例 Keil RealView MDK 开发环境安装在 D 盘, 所以, TKScope 仿真器的驱动安装路径为 D:\Keil, 如图 2.2 所示。

Folder Selection			KSec	BO®
Select the folder where S	ETUP will install files.			he
TKScope SETUP will inst	all in the following folder.			
To install to this folder, pre folder.	ss 'Next'. To install to a differe	t folder, press 'Brow	se' and select ar	other
To install to this folder, pre folder.	ss "Next". To install to a differe	t folder, press 'Brow	se' and select ar	iother
To install to this folder, pre folder. Destination Folder	ss 'Next'. To install to a differe	t folder, press 'Brow	se' and select ar	other
To install to this folder, pre folder. Destination Folder	ss 'Next'. To install to a differe	t folder, press 'Brow	se' and select ar	other
To install to this folder, pre folder. Destination Folder	ss 'Next'. To install to a differe	t folder, press 'Brow	se' and select ar	other

图 2.2 指定驱动安装路径

至此,TKScope 仿真器所需的驱动全部安装完成。在安装目录下(本文示例为 D:\Keil\TKScope),可以看到各个环境下的.dll 驱动文件,如图 2.4 所示。

🚔 IKScope	
文件(E) 編輯(E) 查看(E) 收藏(L) 工具(E) 帮助(B)	1
🔇 GAE - 🔘 - 🎓 🔎 NR  🌔 文件夫 🔟 -	
1812 (D) 😋 D: \Keil\THScope	🗸 🔁 幹到
文件和文件未代考 ④ 信仰 - 小紅文件未 ● 信心 - 文件未完考問 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」	
現完後晋 (A) ULZAHM TISCP DEV ARM for AGUI. dll	
<ul> <li>2×11</li> <li>3×05次倍</li> <li>3×059年間</li> <li>3×059年間</li> <li>3×059年間</li> </ul>	
Haith 😮	

图 2.4 驱动安装目录文件



各个驱动文件的所属类型以及应用的开发环境,用户请详见安装目录下的 readme.txt 文件(本文示例图 2.4 中)。 表 2.1 列举出目前所有的驱动文件,如若驱动文件有所增减或变动,以 readme.txt 文件为准。

## 表 2.1 驱动文件列表

驱动名称	驱动类型	应用环境
UL2ARM_TKSCP_DRV_ARM_for_AGDI.dll	ARM	在 Keil Uvsion4/Uvsion3/Uvsion2 下的驱动
TKSCP_DRV_ARM_for_IAR_v4.dll	ARM	在 IAR V4 版本下的驱动
TKSCP_DRV_ARM_for_IAR_v5.dll	ARM	在 IAR V5 版本下的驱动
TKSCP_DRV_for_RDI.dll	ARM	在 AXD(ADS)下的驱动, 以及其它 RDI 协议

# 2.2 硬件连接

用户在使用 TKScope 仿真器进行仿真之前,需要把仿真器和 PC 机、用户目标板这些硬件设备正确的连接起来。

# 连接仿真器和计算机

TKScope 仿真器通过 USB 接口与计算机连接,即插即用。

# 连接仿真器和目标板

TKScope 仿真器通过 POD-JTAG-ARM-DP20 仿真头与目标板 JTAG 接口连接,支持热插拔。

# TKScope 仿真器上电

用出厂所配的电源适配器给仿真器上电,电源开关拨到 ON 位置,电源指示灯 PWR 呈红色点亮状态,CLK、BSY、ERR 交替闪烁 数次后,BSY、ERR 熄灭,CLK 继续闪烁,之后 USB 指示灯呈蓝色点亮状态,此时仿真器就可以正常的工作了。

注意!必须使用出厂所配的电源适配器,其他规格的电源适配器可能会导致仿真器损坏。



USB 设备第一次上电使用,系统一般都会要求安装 USB 新硬件驱动,否则,系统可能会无法识别到 USB 设备!

1

TKScope 仿真器第一次上电使用,系统会弹出如图 2.5 所示的新硬件安装向导对话框。 此时,需要指定 USB 设备驱动的具体位置。



图 2.5 新硬件安装向导

选择图 2.5 中的【从列表或指定位置安装(高级)】选项,然后点击【下一步】,此时系统会弹出如图 2.6 所示的对话框。

找到新的硬件向导
请选择您的搜索和安装选项。
<ul> <li>● 在这些位置上搜索最佳驱动程序(S)。</li> <li>使用下列的互连框限制或扩展默认搜索,包括本机路径和可移动媒体。会安装找到的最佳驱动程序。</li> <li>● 搜索可移动媒体(软盘、CD-EOM)@)</li> <li>● 在搜索中包括这个位置(D):</li> <li>[]:\keil\TKScope\Driver\TKScope K Driver ● 浏览(B)</li> <li>● 不要推索,我要自己选择要安装的驱动程序(D)。</li> <li>法答这个选项以便从列表中选择设备驱动程序,Windows 不能保证您所选择的驱动程序与您的硬件最匹配。</li> </ul>
(<上ー步 @) (下一步 @) >

图 2.6 选择驱动提示框

3

点击图 2.6 中的【浏览】选项,进入如图 2.7 所示的界面。按照 TKScope 仿真器驱动安装的路径找到驱动文件(本文示例为 D:\Keil\TKScope\Driver\TKScope K Driver),然后点击【确定】。

浏览文件夹	? 🛽			
选择包含您的硬	件的驱动程序的文件夹。			
	🖃 🛅 Driver 🔥			
	표 🚞 AK100 Driver			
	표 🚞 CK100 Driver			
	🚞 TKS BU Driver			
	표 🚞 TKScope K9 DK3 DK5 DK9 🔲			
	🖃 🚞 TKScope K Driver 🧮			
	TKScopeK			
	🗁 WinXP 🗸 🗸			
<				
,				
要查看任何子文件夹,诸单击上面的 + 号。				
	确定 取消			

图 2.7 指定驱动具体位置

指定正确的驱动之后,系统自动进行安装,安装过程中 会弹出如图 2.8 所示的对话框。此时,选择【仍然继续】 选项,让安装过程继续进行。

硬件安装	ŧ
1	正在为此硬件安装的软件: ZHIYUAN TKScope K series Universal Develope Tools 没有通过 Windows 散标测试,无法验证它同 Windows XP 的相容性。(音话我为什么这个测试很重要。) 继续安装此软件会立即或在UL内使无线变得不稳定。
	Blerosott 狂以怒魂在停止其实袭,并同戰件供应尚 联系,以获得通过 Tindows 激标测试的软件。
	仍然继续 (2) 停止安装 (3)

图 2.8 硬件安装过程

驱动安装完毕,系统会弹出如图 2.9 所示的对话框,提 示用户已经完成驱动的安装。此时,点击【完成】即可。 至此,驱动程序安装完毕。



图 2.9 新硬件安装完成



系统正确安装驱动后,可以通过查看设备管理器看到 当前的硬件设备。使用鼠标右键点击【我的电脑】, 选择【属性】,进入如图 2.10 所示的界面。

系统属性 ? 🛛
常规 计算机名 硬件 高级 自动更新 远程
设备管理器 设备管理器列出所有安装在计算机上的硬件设备。请使 用设备管理器来更改设备的属性。
设备管理器 印
- 驱动程序
Windows Update 党庁的方式。     Windows Update 党庁的方式。
驱动程序签名 (3)    Windows Update (2)
硬件配置文件
硬件配置文件向您提供建立和保存不同硬件配置的方 法。
硬件配置文件 (2)
<b>确定 取消</b> 应用 (A)

图 2.10 系统属性

如果系统没有安装新硬件的驱动或驱动安装不正确,那 么会看到如图 2.12 所示的界面。系统检测到新的 USB 设备,但是没有找到正确的驱动,无法正常使用。

文件(E) 操作(A) 查看(V) 帮助(H)	
<ul> <li>□ LUTING</li> <li>□ DIE ATA/ATAFI 控制器</li> <li>□ DIE ATA/ATAFI 控制</li> <li>□ DIE ATA/ATAFI PARA</li> <li>□ DIE ATA/ATAFI PARA<!--</td--><td></td></li></ul>	

图 2.12 没有正确的安装新硬件结果

在图 2.10 中,点击【设备管理器】,进入如图 2.11 所 示的界面。此时,可以在【通用串行总线控制器】一 栏内看到系统识别到的新安装的硬件设备。



图 2.11 正确的安装新硬件结果

无法使用的 USB 设备需要重新安装驱动程序。点击鼠 标右键,选择【更新驱动程序】选项,如图 2.13 所示。 按照上述的过程重新安装驱动程序直到正确为止。



图 2.13 更新驱动程序

# 第3章 Keil RealView MDK 环境 仿真方法





# 3.1 仿真环境设置

点击图 3.1 中的 💦 图标,进入如图 3.2 所示的工程 在 Keil RealView MDK 环境下打开一个编译 OK 的工程, 设置界面。 如图 3.1 所示。 ptions for Target 'Debug\_in\_ChipFlash' Device Target Dubput Liding User CC++ Aun Lidiar Debug Ublice ∩ Lina Senderar Settings Older ThScore Debug to ABM V Settings Eile Edit Fiew Project Debug Flash Peripherals Lools SWCS Mindow Melp O Use Simulator 🖄 🥔 🖬 🕼 👗 🐘 🖄 **× M** /4 保水外为防锅 hipFlash V A To The Guargerout + 🐚 🚳 🍳 📧 🎮 🗄 🌚 🖾 Load Application at Statup
 Plun to main()
Initialization File: Load Application at Startup Initialization File: 🕸 🗉 🕬 Ă 🛛 🙀 🕵 Debug\_in\_ChipFlash Mebug\_in\_DhipFLASH.ini ... [d... .. Edt.. orkspace Guangzou 2LG-MCU . graduate s http://www.zlga 01 0 02 03 Restore Debug Session Settings V Breakpoints V Toolbox V Watchpoints V Memory Display Breakpoints
 Watchpoints LPA
 Wemory Display • 04 05 06 07 08 09 10 11 
 06
 \*\*
 File Info

 07
 \*\* File name:
 main.c

 08
 \* Last point parts: 000:09-16

 10
 \* Description:

 11
 \*\*

 12
 \*\*

 14
 \* Created by:

 15
 \* Version:

 16
 \* Description:

 16
 \*\*

 17
 \*\*
 CPU DLL: Parameter: SARM.DLL <cLPC2100 Driver DLL: Parameter: SARM.DLL Dialog DLL: Parameter. DARMP.DLL pLPC2132 Dialog DLL: Parameter: TARMP.DLL pLPC2132 . ū 🖹 main e · 确定 取消 Default: 帮助 #111-D: stat nking... ogram Size: Code=1764 RO-data=16 RW-data=0 ZI-data=1124 \Project\_Data\Debug\_in\_ChipFlash\Project.axf\* - 0 Error(s), C\_D\_D\**budd** <u>{</u> Command <u>}</u> EndinFles <u>/</u>

图 3.2 工程设置界面

图 3.1 MDK 主界面

1 >

在图 3.2 中,选择硬件仿真,对应的驱动选择【TKScope Debug for ARM】。然后,点击【Settings】进入 TKScope 仿真器设 置界面,如图 3.3 所示。

设置为 K9   POD-JTAG-ARE	-P20   LPC2132   NKP	
硬件选择	(1) 厂商: NXP	<u>^</u>
主要设置	(2)器件: LPC2132 (3) 仿真器: K9 (4) POP 第型: POP ITAC ADM P2P	
TAP设置	(1) FOO 安全: FOO JIASAAN #20 目总症小框 主要设置	
程序烧写	(1) 緩沖代码。 (2) 緩冲数層。 (2) 歩端	
初始化宏	(4)使用硬件复位. (5)复位保持延时:50毫秒.	
硬件自检	(6)复位恢复延时: 500毫秒. (7)软件复位. (8)自动停止	
	(9) 系统时钟: 35.5000MHz.	~
	加載(保存)、缺省 職以 取消	搜索

图 3.3 仿真器设置界面

图 3.3 中, 点击左侧的各个选项, 系统会弹出相应的设置界面, 同时右侧的信息提示框中会出现各项设置信息的具体含义。

# 3.2 仿真器参数设置



TKScope 仿真器工作参数必须要正确设置,否则可能会导致仿真错误或失败!

# 3.2.1 硬件选择

点击图 3.3 中的【硬件选择】,进入如图 3.4 所示的界面。

仿真器类型 POD类型		器件信息	器件过滤	
K9 POD-JTAG-ARM-P20	~	LPC2132 / NXP		
		1.仿真芯片: LPC2132 / NXP LPC2131 LPC2132 LPC2136 LPC2136 LPC2138 * 16/32位ARM7TDMI-5核,超小UQFP64; * 036(5328片内静态RAM和32/64/128/2 * 126位宽度接行)加速器可实现高达60 * 通过片内bock装载程序实现在系统编 * 单个目sh点层区或整片描绘时因为400 * EmbeddedICE RT和嵌入式跟踪接口,	封装. 55(51218月内Flash. 附4工作频率. 程/在应用编程(ISP/IAP). ns。256字节行编程时间为1ms.	

图 3.4 硬件选择界面





用户可以直接选中仿真芯片型号和当前使用的仿真器种类。

友情提示:用户可以在【器件过滤】框内输入器件的名称,系统会自动帮助您快速找到器件。

用户也可以利用系统提供的【搜索】功能来完成。具体方法:用户只需选中仿真芯片型号,点击【确定】返回到图 3.3 的界面; 然后点击图 3.3 中的【搜索】,系统会自动搜索出当前连接的所有仿真器和仿真头信息,如图 3.5 所示。 如果您的电脑连接多个仿真器,需要选择您当前使用的那台仿真器,然后点击【确定】。

设备列表				
仿真器	串号	LAB	POD	确定
К8	00000000016	LAB-1064-256K	POD-JTAG-ARM-DP20	取消
🗹 К9	000000001110	LAB-1064-512K	POD-JTAG-ARM-DP20	

图 3.5 系统搜索结果

# 3.2.2 主要设置

点击图 3.3 中的【主要设置】,进入如图 3.6 所示的界面, 用户根据仿真需要设置即可。

主要设置	<
cache ☑ 缓冲代码	
单步 ☑ 使用软件单步	
☑ 使用软件断点	
Endian ④小端 〇大端	
时钟 学校 25,0000	
时钟模式 ○ 自动时钟 ● 同步时钟 ● 固定时钟	
硬件复位 50 ms复位保持时间 √系统复位 ✓ Jtag复位 100 ms复位恢复时间	
操作策略	
内核复位 内核停止 不复位. ▼ 自动停止. ▼	
<b>適定</b> 取消	

图 3.6 主要设置界面

#### 显示缓存配置

缓存配置(Cache)是解决屏幕刷新和仿真速度的矛盾。 如果**选择** Cache,屏幕显示刷新只在用户程序运行后进 行,可加快显示速度。但是,如果某一个操作引起的其 它数据的变化可能不能及时显示。

如果**不选择** Cache,则用户在 PC 端的任何操作都将引 起显示数据的重新刷新。在查找不稳定硬件时比较理想, 但是屏幕刷新会影响操作响应速度。

【缓存代码】: 用户代码缓存。

【缓存数据】: 非用户代码缓存。

建议用户使用缺省配置,仅选择【缓存代码】。

# 单步

【使用软件单步】:选中此项,则 Flash 断点不会重复 编写,加快 Flash 断点调试速度。

# 断点

【使用软件断点】: 选中此项,则可实现 RAM 中无限 制断点调试。

【使用 Flash 断点】:选中此项,则可实现 Flash 中无限制断点调试。



h

# 大小端选择

选择当前存储器模式为大端或小端。如果当前 MCU 为 固定的大小端模式,则该选项无法由用户选择。

# 时钟频率选择

【系统时钟】:用户系统最终的运行频率,在 Flash 编 程时使用,单位 MHz。

【JTAG 时钟】:选择 JTAG 时钟频率,仅在固定时钟 下有效。

# JTAG 时钟模式选择

【自动时钟】: 自动选择复位后最高的可用时钟。 【同步时钟】: 根据目标板返回的同步时钟速度,选择 最佳时钟。如果目标板上没有同步时钟输入,选择后将 会运行的非常缓慢。

【固定时钟】:选择用户输入的时钟频率数值。



【系统复位】: 使用硬件复位 nSRST。 【JTAG 复位】: 使用硬件复位 nTRST。 【复位保持时间】:选择复位有效时期的延迟时间,单位 ms。 【复位恢复时间】:选择复位结束时期的延迟时间,单位 ms。



图 3.7 硬件复位信号图



复位恢复时间值的大小取决于用户目标板上的复位器件参数,时间值的设置应大于复位器件的复位时间,否则,可能会 出现系统没有完全复位而仿真器却开始进入调试状态的情况,最终导致仿真失败!

# 操作策略

【内核复位】和【内核停止】选项共同构成停止策略,满足客户的各种不同调试需要。 【不加复位,软件停止】:不施加硬件复位,使用软件停止的方法。 【可能复位,自动停止】:可能施加硬件复位,采用一切可能的停止方法。 【施加复位,DBRQ 停止】:施加硬件复位,使用 DBRQ 信号停止。 【施加复位,断点停止】:施加硬件复位,使用断点停止。 【施加复位,特殊停止】:施加硬件复位,根据不同芯片采用不同的停止方法。

# 3.2.3 TAP 设置

点击图 3.3 中的【TAP 设置】,进入如图 3.8 所示的界面。



图 3.8 TAP 设置界面

TAP 设置选项用于设置 JTAG 链的器件参数,包括器件 个数、顺序、IR 长度、当前仿真器件。

当扫描链中包含未知的器件或存在多个可以仿真的器件 时,用户必须进行设置,设置时尽量参考自动生成的扫 描链参数 (可以应付绝大多数的情况)。

【器件列表】: 扫描链中存在器件的参数,包括 IDCODE、器件名称、IR 长度。 【IDCODE】: 输入的器件 IDCODE 数值(16 进制),用于添加或更新器件。 【器件名称】: 输入的器件名称,可仿真的器件必须包含"ARM"字符串以供识别。 【IR 长度】: 器件的 IR 长度,数值不能为 0。 【自动检测】: 根据已登记的器件自动配置扫描链,不能完成时启动人工配置。 【人工配置】: 每次配置前启动该配置窗口,用户手动选择。 【太加】: 在器件列表中添加一个新的器件,IDCODE、器件名称、IR 长度必须指定。 【删除】: 在器件列表中删除当前器件。 【更新】: 更新器件列表中当前器件的参数。 【上升】: 在器件列表中,将当前器件的位置上移。 【下降】: 在器件列表中,将当前器件的位置下移。

# 3.2.4 程序烧写

点击图 3.3 中的【程序烧写】,进入如图 3.9 所示的界面。

程序烧写			$\mathbf{X}$
编程选项 ○整片擦除 ③ 扇区擦除 ○ 不予擦除	✔ 编程Flash ✔ 验证Flash □ 代码相同跳过	·装载算法RA 起始 0x40	M 000000 尺寸0x00003FE0
编程算法 编程描述 LPC2000 IAP 256kB Flash	器件类型 ONCHIP	器件尺寸 0×0003E000	地址范围 0x00000000 - 0x0003DFFF
	添加算法	起始 0×00	000000 尺寸 0×0003E000 确认 取消

图 3.9 程序烧写界面

# Flash 编程算法

【起始】:当前算法文件的偏移地址。 【尺寸】:当前算法文件的用户修改尺寸,不能大于原始文件中的尺寸。

该区域显示的为当前加载的 Flash 编程算法文件,可同时加载多个,但地址不能重叠。用户可根据随机文档中提供的算法 文件范例进行编写,尤其是外部 Flash 的烧写。

Flash 编程设置

装载算法 RAM

减小可加快下载速度。

【不予擦除】:不进行 Flash 擦除。

【整片擦除】: 程序下载到 Flash 前擦除全部 Flash 空间。 【扇区擦除】: 根据下载需要擦除相应的扇区空间。

【起始】: 可用的 RAM 起始地址。用户一般不需要修改。 【尺寸】: 当前 RAM 的尺寸,给定的数值满足需要。适当

## 编程算法文件的操作

【添加算法】:添加用户指定的编程算法文件。 【删除算法】:删除当前选择的算法文件,内部 Flash 的算法文件不能删除。



用户如果选择在 Flash 中调试,必须选中【编程 Flash】、【验证 Flash】选项,同时选择【整片擦除】或【扇区擦除】。 另外,建议用户选中【代码相同跳过】选项,这样对于重复编程 Flash 的情况,可以大大提高编程速度。

【程序烧写】是比较重要的一项设置,直接关系到仿真能否成功以及仿真结果是否正确。下面将详细讲解几种常用的 Flash 编程烧写 方法。

## 芯片内部 Flash

用户如果仿真内部带有 Flash 的芯片,系统会自动调用芯片的 Flash 算法文件。此时,选中【编程 Flash】、【验证 Flash】选项, 同时选择【整片擦除】或【扇区擦除】即可。

例如,NXP LPC2132 芯片内部带有 64K Flash,选择在此芯片内部 Flash 调试,设置界面如图 3.10 所示。

程序焼写				X
編程选项 ○ 整片擦除 ○ 扇区擦除 ○ 不予擦除	✔ 编程Flash ✔ 验证Flash ✔ 代码相同跳过	装载算法R/ 起始 0x4	am 0000000 尺寸 0x00001FE0	
编程描述 LPC2000 IAP 256kB Flash	器件类型 ONCHIP	器件尺寸 0×0003E000	地址范围 0x00000000 - 0x0003DFFF	
	(注: ) 二、 ([::::::) (∶: ) 二、 ([::::::) (∶: ) 二、 ([::::::) (∶: ) (∶: ) 二、 ([::::::) (∶: ) (∶	起始 0×0	0000000 尺寸 0x0003E000	
	添加算法	「「「「「」」「「」」「「」」「」「」」「」」「」」「」」「」」「」」「」」「	明认 取消	

图 3.10 程序烧写界面 1

# 外部扩展 Flash

用户如果仿真内部无 Flash 的芯片,需要把程序下载到外部扩展的 Flash 中。此时,需要添加外部 Flash 的算法文件,并正确 设置 Flash 的起始地址和尺寸大小。

例如, NXP LPC2220 芯片内部无 Flash, 外部扩展 SST39VF160, 选择在外部 Flash 中调试。点击【添加算法】, 把 SST39VF160 的算法文件添加进来, 起始地址根据 LPC2220 芯片特性设置为 0x80000000, 设置界面如图 3.11 所示。

程序集写 編程选项 ○ 整片接除 ○ 扇区接除 ○ 示子接除	<ul> <li>※載算法RAM</li> <li>※載算法RAM</li> <li>◇ 编程Flash</li> <li>2 验证Flash</li> <li>2 代码相同跳过</li> </ul>	
编程描述 编程描述 SST39×160x Flash	器件类型 器件尺寸 地址范围 EXT16BIT 0×00200000 0×80000000 - 0×801FFFFF	
	超始 0x80000000 尺寸 0x00200000	
	[一添加算法] 一册除算法 确认 取消	

图 3.11 程序烧写界面 2

TKScope 仿真器提供众多外部 Flash 的算法文件,存放路径为 TKScope 驱动安装目录下 TKScope\configuration\ExtFlash。

# 芯片内部 Flash+外部扩展 Flash

用户如果使用内部带有 Flash 的芯片,同时系统外扩 Flash 空间,这种情况需要把程序顺次下载到内部 Flash 和外部 Flash 中。 芯片内部 Flash 的算法文件,系统会自动调用。外扩 Flash 的算法文件,用户需要手动添加进来,添加方法同上,可同时加载 多个 Flash 算法文件。

此时,需要特别注意各个 Flash 的起始地址和尺寸大小的设置,地址不能重叠。

例如,NXP LPC2292 芯片内部带有 256K Flash,外部扩展 2 片 SST36VF1601,选择芯片内部 Flash 和外扩 Flash 联合仿真 的模式。LPC2292 的算法文件,系统会自动调用; SST36VF1601 的算法文件,用户需要手动添加进来。然后,根据 LPC2292 芯片特性,外扩 Flash 的起始地址分别设置为 0x8000000、0x81000000,设置完毕之后检查各个 Flash 无重叠现象,界面如 图 3.12 所示。



#### 图 3.12 程序烧写界面 3

# RAM 调试

用户如果选择在 RAM 中调试,无需选中【编程 Flash】、【验证 Flash】选项,同时选择【不予擦除】,界面设置如图 3.13 所示。

置 <mark>序烧写</mark> 。 《编程选项		—————————————————————————————————————	M	2
<ul> <li>整片擦除</li> <li>扇区擦除</li> <li>豖子擦除</li> </ul>	<ul> <li>■ 編程Flash</li> <li>■ 验证Flash</li> <li>■ 代码相同跳过</li> </ul>	起始 0x40	000000 尺寸[0x00003FE0	]
编程算法				_
编程描述	器件类型	器件尺寸	地址范围	
LPC2000 IAP2 64kB Flash	ONCHIP	0×00010000	0x00000000 - 0x0000FFFF	
		起始 0×00	000000 尺寸 0x00010000	]
	添加算法	删除算法	()	

图 3.13 程序烧写界面 4

# 3.2.5 初始化宏

点击图 3.3 中的【初始化宏】,进入如图 3.14 所示的界面。

#### 初始化宏 D:\Keil\TKScope\configuration\LPC2132.ini 确认 取消 备注 MEMMAP VICIntEnClear 序号 类型 0 写32位 1 写32位 参数1 参数2 0×E01FC040 0×1 0×FFFFF014 0×FFFFFFF 参数3 参数4 下降 添加 册除 插入 更新 时期 类型 复位后 ▼ 地址(Hex) 数据(Hex) 0x0 注解

图 3.14 初始化宏界面

用户如果需要设置初始化宏文件,可以在此界面导入,点击 图 3.14 中的 ..... 图标,导入初始化文件即可。



TKScope 仿真器为数百种芯片提供完善的 初始化文件,存放路径为 TKScope 驱动安 装目录 TKScope\configuration\ExtFlash。 这种情况下,系统会自动导入芯片对应的 初始化文件,用户无需自行导入。

初始化宏是后缀为.ini的宏执行文件,用于在不同时期执行不同操作。

#### 表 3.1 现在支持的时期

名称	意义
preRESET	复位前执行的宏操作。
postRESET	复位后执行的宏操作。
preRUN	运行前执行的宏操作。
postRUN	运行后执行的宏操作。
preFlash	写入前执行的宏操作。
postFlash	写入后执行的宏操作。
preDownload	程序下载前执行的宏操作。
postDownload	程序下载后执行的宏操作。

### 表 3.2 现在支持的动作

名称	意义
Read 32bit	读取指定地址的 32 位数据。
Read 16bit	读取指定地址的 16 位数据。
Read 8bit	读取指定地址的 8 位数据。
Write 32bit	在指定地址写入指定的 32 位数据。
Write 16bit	在指定地址写入指定的 16 位数据。
Write 8bit	在指定地址写入指定的8位数据。
SetJtagClock	指定 Jtag 工作模式和工作频率(10 进制)。
Delay	延时一段时间,单位为 ms(10 进制)。
SetPC	设置当前程序指针(16 进制)。
Message	输出字符串信息到 IDE 界面中。



每个时期支持无限制次数的动作,编号从 0 开始。Ini 文件为标准格式,分成不同的节,每个节中有键和键值组成。

下面是一个 Ini 文件范例。用户可根据实际仿真情况,自行编写初始化文件。

[postRESET] InitStep0\_Action = "Write 32bit" InitStep0\_Comment = "MEMMAP internal flash" InitStep0\_Value0 = 0xE01FC040 InitStep0\_Value1 = 0x1

InitStep1\_Action = "SetJtagClock" InitStep1\_Comment = "1MHz 固定时钟" InitStep1\_Value0 = "FixedJtagClock" InitStep1\_Value1 = 10000000

InitStep2\_Action = "Read 32bit" InitStep2\_Comment = "读取 32 位数据" InitStep2\_Value0 = 0xE01FC080 InitStep2\_Value1 = 0

InitStep3\_Action = "Delay" InitStep3\_Comment = "延时 1000ms" InitStep3\_Value0 = 1000 InitStep4\_Action = "Message" InitStep4\_Value0 = "喂! 这是一个信息!"

InitStep5\_Action = "SetPC" InitStep5\_Comment = "设置 pc 指针"

[preRun] InitStep0\_Action = "SetJtagClock" InitStep0\_Comment = "设置自动时钟" InitStep0\_Value0 = "AutoJtagClock"

[postRun] InitStep0\_Action = "SetJtagClock" InitStep0\_Comment = "设置同步时钟" InitStep0\_Value0 = "SycJtagClock"

# 3.2.6 硬件自检

点击图 3.3 中的【硬件自检】,进入如图 3.15 所示的界面。

件自检			
检查 TKScope K8 发现主板,但是在该仿真模式下无法检查, 请选择其它仿真模式进行检查操作。			
检查 L48_1064_256K 发现L48_但是在该逻辑模式下无法检查。 请选择其它逻辑模式进行检查操作。 			3
2871 / 100000. 正确! 复位成功, IDCODE = 0x4F1F0F0F.			~
	开始	助过	结束

图 3.15 硬件自检界面

TKScope 仿真器的硬件自检功能主要是检测硬件初始化、USB 通讯、硬件复位以及 ARM 芯片 ID 的 100000 次读写。

Q			
	$\vee$		
		9	

硬件自检是非常实用的一项功能,可以用来 检测仿真器与计算机、目标板的通讯情况。 用户在使用过程中,遇到联机通信失败的情 况,可以利用硬件自检功能来判断故障产生 原因。

# 3.3 仿真调试

仿真器工作参数设置完成之后,就可以进行仿真了。

# 3.3.1 开始仿真调试

点击【Debug】菜单下的【Start/Stop Debug Session】选项,或点击快捷图标 🙆 ,即可进入仿真调试状态,如图 3.13 所示。



图 3.13 进入调试状态

进入仿真状态之后,用户可以根据实际仿真需要,选择相应的调试工具进行仿真操作。

# 表 3.3 Keil RealView MDK 运行调试工具

快捷图标	意义
Q	进入/退出调试状态(Start/Stop Debug Session)。
RST	复位(Reset CPU)。
	全速运行(Go)。
8	停止运行(Stop)。
7	单步运行(Step In)。与 Step 命令不同之处在于对函数调用语句,Step In 命令将进入该函数。
0	单步运行(Step)。每次执行一条语句,这时函数调用将被作为一条语句执行。
በት	单步运行(Step Out)。执行完当前被调用的函数,停止在函数调用的下一条语句。
*()	运行到光标(Run To Cursor)。程序运行到当前光标所在行时停止。
	设置/取消断点(Insert/Remove Breakpoint)。
1	取消所有断点(Kill All Breakpoints)。
<b>W</b>	打开/关闭断点(Enable/Disable Breakpoint)。
<b>(D</b>	关闭所有断点(Disable All Breakpoints)。

# 表 3.4 Keil RealView MDK 调试观察窗口工具

快捷图标	意义
R	打开反汇编窗口(Disassembly Window)。
<b></b>	打开变量观察窗口(Watch and Call Stack Window)。
	打开存储器观察窗口(Memory Window)。

# 3.3.3 仿真调试结果

用户仿真结束,点击【Debug】菜单下的【Start/Stop Debug Session】选项,或点击快捷图标 🔍 ,即可退出仿真调试状态。



用户如果选择在 RAM 中调试,程序写到芯片的 RAM 中,掉电后会丢失,再次上电目标板不会执行相应的程序。



用户如果选择在 Flash 中调试,程序写到芯片的 Flash 中,掉电后会保存,再次上电目标板会执行写入的程序。



用户如果选择带有加密配置的 Flash 中调试,程序写到芯片的 Flash 中,而且已经加密,再次上电目标板会执行写入的 程序。但是,此时不能使用任何调试工具再次进行仿真调试。除非,使用 ISP 软件全局擦除芯片,才可再次进行仿真 调试。此种方式用于用户烧写最终的程序到芯片中,而且芯片已经被加密。



# 第 4 章 ADS 环境仿真方法

添加驱动文件		28
仿真调	试	29
4.2.1	仿真调试工具	29
4.2.2	仿真调试结果	30
	添加驱 仿真调 4.2.1 4.2.2	<ul> <li>添加驱动文件</li> <li>仿真调试</li> <li>4.2.1 仿真调试工具</li> <li>4.2.2 仿真调试结果</li> </ul>



在 ADS 环境下打开一个编译 OK 的工程,如图 4.1 所示。点击图中 💽 图标,即可进入 AXD 调试环境, 如图 4.2 所示。

Letrowerks CodeWarrior for ARM Developer Sui	te vl.	2 [	
<u>File E</u> dit <u>V</u> iew <u>S</u> earch <u>P</u> roject <u>D</u> ebug <u>W</u> indow <u>H</u> elp			- 8 :
<ul> <li>DebugInFLASH</li> <li>DebugInFLASH</li> </ul>	<b>i</b> \$	:	
Files Link Order   Targets			
File	Code	Data 🐐	2. 🐇
⊡ 🔄 sef	0	0	• 🔳
mem_c.scf	n/a	n/a	• 🔳
📲 mem_a. scf	n/a	n/a	• 🔳
mem_b.scf	n/a	n/a	• =
E Cath	0	0	• =
target.h	0	0	· =
Config. h	E16	112	·
Startun s	288	1028	8
Torget c	228	1020	115
IR9. s	0	ŏ	
user	144	4	
main.c	144	4	• • <del>-</del>

图 4.1 ADS 主界面

 在 AXD 调试环境下,选择【Options】菜单下

 【Configure Target】选项,如图 4.2 所示。



图 4.2 AXD 主界面

3

系统弹出 Choose Target 窗口,如图 4.3 所示。点击图 中【Add】按钮,添加 TKScope 仿真器的驱动文件。





系统弹出驱动选择对话框,打开 TKScope 仿真器驱动安 装目录(本文示例安装路径为 D:\Keil\TKScope),选中 TKSCP\_DRV\_for\_RDI.dll,如图 4.4 所示。

11.71			
查找范围(I):	TKScope		
🚞 configura	tion	TKSCP_DRV_for_RDI. dll	
🛅 Driver		SUL2ARM_TKSCP_DRV_ARM_for_	AGI
ProgConfi	g. dll		
TKSCP_DRV	ALL_for_AGDI. dll		
TKSCP_DRV	_ARM_for_IAR_v4. dll		
STKSCP_DRV	ARM_for_IAR_v5. dll		
<	iii j		>
文件名(图):	TKSCP_DRV_for_RDI. dll	打开 (0)	
	-		

图 4.4 选择驱动文件

驱动安装完成之后, Choose Target 窗口会显示当前安 装的驱动选项,如图 4.5 所示。选中 TKScope 仿真器的 驱动,点击【Configure】按钮,即可进入 TKScope 仿 真器设置界面,如图 4.6 所示。

Cho	ose Tar	get		? 🗙
_ 1	Farget Env	ironments		
]	Farget	RDI	File	Add
	ADP ARMUL TEScore	1.5.1 1.5.1	D:\\Bin\Remote_A.dll d:\PROGRA~1\\Bin\ARMulate.dll D:\Keil\ KSCP_DBV_for_BUL_dll	Remove
1				Rename
				Save As
3	(			Configure
TK	Scope ARM	Tools set for A	DS, V1.06, 2008/01	
			OK Cancel	Help

图 4.5 驱动安装完毕

无论在哪种 IDE 环境下,TKScope 仿真器设置界面都是一样的(例如,ADS 环境下的图 4.6 与 Keil RealView MDK 环境下的 图 3.3),设置方法也是一样的。



图 4.6 仿真器设置界面



TKScope 仿真器工作参数必须要正确设置, 否则可能会导致仿真错误或失败! 具体设置方法用户请参考 3.2 小节《仿真器参数设置》,这里不再重复叙述。

# 4.2 仿真调试

在 AXD 调试环境下,安装驱动并正确设置 TKScope 工作参数之后,就可以进行仿真调试了。

# 4.2.1 仿真调试工具

## 表 4.1 AXD 运行调试工具

快捷图标	意义
	全速运行(Go)。
E	停止运行(Stop)。
7	单步运行(Step In)。与 Step 命令不同之处在于对函数调用语句,Step In 命令将进入该函数。
Ð	单步运行(Step)。每次执行一条语句,这时函数调用将被作为一条语句执行。
ዮ	单步运行(Step Out)。执行完当前被调用的函数,停止在函数调用的下一条语句。
*0	运行到光标(Run To Cursor)。运行程序直到当前光标所在行时停止。
Ē	设置断点(Toggle BreakPoint)。

# 表 4.2 AXD 调试观察窗口工具

快捷图标	意义
r	打开寄存器窗口(Processor Registers)。
	打开观察窗口(Processor Watch)。
V	打开变量观察窗口(Context Variable)。
	打开存储器观察窗口(Memory)。
Q	打开反汇编窗口(Disassembly)。

# 表 4.3 AXD 文件操作工具

快捷图标	意义
100 100	加载调试文件(Load Image)。
¢	重新加载文件(Reload Current Image)。由于 AXD 没有复位命令,所以通常使用 Reload 实现复位(直接更 改 PC 寄存器为零也能实现复位)。

# 4.2.2 仿真调试结果

用户仿真结束,直接关闭 AXD 调试环境即可退出仿真调试状态。



用户如果选择在 RAM 中调试,程序写到芯片的 RAM 中,掉电后会丢失,再次上电目标板不会执行相应的程序。



用户如果选择在 Flash 中调试,程序写到芯片的 Flash 中,掉电后会保存,再次上电目标板会执行写入的程序。



用户如果选择带有加密配置的 Flash 中调试,程序写到芯片的 Flash 中,而且已经加密,再次上电目标板会执行写入的 程序。但是,此时不能使用任何调试工具再次进行仿真调试。除非,使用 ISP 软件全局擦除芯片,才可再次进行仿真 调试。此种方式用于用户烧写最终的程序到芯片中,而且芯片已经被加密。

# 第 5 章 IAR 环境仿真方法



5.1	添加驱动文件		32
5.2	仿真调	试	34
	5.2.1	开始仿真调试	34
	5.2.2	仿真调试工具	35
	5.2.3	仿真调试结果	36



在 IAR 环境下打开一个编译 OK 的工程。选中工程, 点击鼠标右键,在弹出的菜单中选择【Options】选 项,如图 5.1 所示。

🔀 IAR Embedded Vor	kbench IDE	
<u>File Edit View Projec</u>	rt Ioslz ≝indow Help	
🗅 🚅 🖬 🕼 🖉	B B 10 00	■ ペ > > > 回 > → ↔ ☆ □
Workspace	× demo.c mervu.c	* x
Debug flash	•	-
Files	ta Big /********	
E Lpc213x_de	Options	Name: main rs: void
- Common So.	llake	void
- demo.c	Compile	ion: Main subroutine
- I user_tunc.c	Rehuild All	
-e ouput	Ciean	wid)
	Stop Build	
	àdd 🕨	ConcernSaver - MENU LON FORTE DLY:
	Renove	LightAutoOffCount = 0;
	Sogree Code Control +	<pre>&gt;a, TranBuff(30); e_t CurrTime;</pre>
	File Properties	e_t CurrData;
	Sgt as Active	t() == 0)
Lpc213k_demo_	140 141	
* Messages		~
Total number of e	mors: 0	
Total number of v	ramings: 0	~
		×
Build Debug Log		×
Edit options for the sel	ected item	Errors 0, Varnings 0

图 5.1 IAR 主界面

在工程配置界面,选择【Debugger】选项,右侧的 【Setup】窗口设置如图 5.2 所示。【Driver】选择 【Third-Party Driver】,选中【Run to main】。

Options for node	"Lpc213x_demo"
Category: General Options C/C++ Compiler Assembler Duild Actions Inder Cetuager Simulator Angel IAR ROM-monitor J-LR(A)-Trace LVII FTOI Macraigor RDI Third-Party Driver	Factory Settings Setup Download   Extra Options   Flagins    Priver Third+Party Driver Satup macros Use macro file FFD0_DIENconfig(Flash mac Derige description file FFD0_DIENconfig(Flash mac Derige description file FFD0_DIENconfig(Flash mac
	OK. Cancel

图 5.2 Debugger Setup 界面

图 5.2 中点击【Download】选项,设置界面如图 5.3 所示,所有的选项均不选中。

C/C++ Compiler Setup Download Extra Options   Plugins   Assembler   Attach to progg
Sundarions Didar Celugar Smplor Angel JaR ROM-monitor J-link/J-Trace LNI FTDI Marcigor RDI

图 5.3 Debugger Download 界面



选择【Third-Party Driver】选项,界面如图 5.4 所示。 点击图中 ... 图标,添加 TKScope 仿真器驱动。

Options for node	"Lpc213x_demo"	×
Category: General Options C/C++ Compiler Assembler	Factory Settings	
Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-monitor J-Link/J-Trace	LAK debugger driver	
LMI FTDI Macraigor RDI Third-Party Driver	Log generation     STOOLLIT_DIRS\cryptome.log     OK Cencel	

图 5.4 Third-Party Driver 界面

5

系统弹出驱动选择对话框,打开TKScope仿真器驱 动安装目录(本文示例安装路径为D:\Keil\TKScope)。 如果用户使用的是IAR V4版本的软件,选择 TKSCP\_DRV\_ARM\_for\_IAR\_v4.dll; 如果用户使用的是IAR V5版本的软件,选择 TKSCP\_DRV\_ARM\_for\_IAR\_v5.dll。 本文示例使用的是IAR V4.42,选择如图 5.5 所示。



图 5.5 选择驱动文件

6

驱动安装完成之后, Third-Party Driver 界面会显示当前安装的驱动选项, 如图 5.6 所示。

Options for node	"Lpc213x_demo"		×
Options for node Category: General Options C/C++ Compler Assembler Custom Build Build Actions Linker Debugger Simulator Angel JAR ROM-monitor bill MC/Troze	<pre>~ Ipr:213z_demo~ Third-Ferty Driver IAR debugger driver [D:\Keil\TEScope\TESCP_DEV_ASM_for_IAR_v4.dll</pre>	Factory Settings	
PDI Third-Party Driver	Log generation     STOULAIT_DIRE\cspyceme.log     OK	Cancel	

图 5.6 驱动安装完毕

▶ 驱动安装完成之后, IAR 主界面菜单栏会显示 【TKScope】选项,如图 5.7 所示。

Classical Xiew Project Tables (Loberto Control	K Embedded Vorkbench	IDE	
Common So.     C	Edit View Project TEScope	Icols Mindow Help	
Workshold     Image: Solution of the solution of th	🛎 🖬 🕼 🖉 🖓 🕹 🛍	□ □ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	***
Debug tish       • Procession Neares waters         Files       • Procession Neares waters         • Uncelling Neares waters       • Variable staters         • If Counting Neares       • Output         • If Counting Neares       • Outp	pace ×	demo.c menu.c	
Messages Total number of errors: 0 Total number of warnings: 0	ag fash v 3 2 200 re 3 2 200 1) Common So. 1) Common So. 1) Modul Source 1) demo e 1) demo e 1) demo e 1) demo e 1) demo e 2) Output 2) Souther source 2) Souther source 2) Souther source 2) Souther source 2) Souther source 2)	<pre>Provide Name: main Parameters void baturn: void Description: Main subroutine void main (void) ( // // NTHO Temp: LPC_INTHO TEMP: LPC_INT</pre>	
Total number of errors: 0 Total number of warnings: 0 K	Messages		
<	Total number of errors: 0 Total number of warnings: 0		
			2
Build Debug Log	d Debug Log		

图 5.7 驱动安装完成主界面



选择【TKScope】菜单下的【Setup】选项,如图 5.8 所示,即可进入 TKScope 仿真器设置界面。

X IAK Embedded Vorkbench IDE		×
Eile Edit Yiew Project INScope Locas Mindow Melp		
	a 64	
Workspace		- 14
Datus Bach		=
		-
Files 22 Mg * Punction Neuron main		
B Lpc213x_de V Parameters: void		
Common So # Return: void		
He Modul Source		
Description: Main subroutine		
La gloser_onece		
woid sain (woid)		
l l		
LFC_INT8U Temp;		
LFC_INT16U ScreenSaver = MENU_LOW_POWER_DLY;		_
LPC_INT16U LightAutoOffCount = 0;		
char RecData, TranBuff[30];		
LPC_RCC_TIMe_C_CUPPTIMe;		
Erc_Roo_Pace_C CuttPaca;		
if(SysInit() == 0)		-
Lpc213x_demo 4	•	
× Hoseano		~
Tetal surplus of oness 0		-
Total number of errors, 0		
r orai number or wannings. o		~
	2	
Build Debug Log		
Errors 0, Warnings 0	3	1- 10

图 5.8 选择 TKScope 仿真器设置



图 5.9 仿真器设置界面



TKScope 仿真器工作参数必须要正确设置,否则可能会导致仿真错误或失败! 具体设置方法用户请参考 3.2 小节《仿真器参数设置》,这里不再重复叙述。

# 5.2 仿真调试

仿真器工作参数设置完成之后,就可以进行仿真了。

# 5.2.1 开始仿真调试

点击【Project】菜单下的【Debug】选项,或点击快捷图标 🐒 ,即可进入仿真调试状态,如图 5.10 所示。



图 5.10 进入调试状态

进入仿真状态之后,用户可以根据实际仿真需要,选择相应的调试工具进行仿真操作。

# 表 5.1 IAR 运行调试工具

快捷图标	意义
<u> </u>	编译并调试(Make and Debug)。
٩	设置断点(Toggle Breakpoint)。
	复位(Reset)。
	停止运行(Break)。
Z	布越(Step Over)。单步执行程序,跳过子程序,不进入到内部。
æ	步进(Step Into)。单步执行程序,进入到子程序内部。
£	步出(Step Out)。执行到当前子函数的结束。
<u>**</u>	运行到下一条语句(Next Statement)。程序运行到下一条语句时停止。
Ž	运行到光标(Run to Cursor)。程序运行到当前光标所在行时停止。
<u>***</u>	全速运行(Go)。
2	退出调试(Stop Debugging)。

在仿真调试过程中,用户可以打开【View】菜单下的相应窗口观察仿真调试的结果。如图 5.11 所示,可以打开寄存器窗口,观察各 个寄存器值的变化情况。同样,也可以打开反汇编窗口、存储器窗口、变量观察窗口等等。

🔀 IAR Enb	edded Vorkbench	IDE	
<u>F</u> ile <u>E</u> dit	Yier Project Debug	Disassembly TEScope Tools Window Help	
🗅 🚅 🗐	Hessages +	∞ ∞	* * * II = * & &   01
5 0 2	Torkspace	<b>\$</b>	
Workspace	Ponce prosret	nenu.c demo.c * X	Disassembly 🗙
Debug flash Files	Bregkpoints Disassembly	MenuInit(MENU_ENGLISH_LANG,MENU_	Goto 💌
🗄 💽 Lpcź	Sumbalic Henory	RTC_Enable();	Bain:
	Begister	<b>)</b> ′	00001F4A B08B LPC_INT16U ScreenSe
- C de	Yatch '0 Locals	* Function Name: main	00001F4C 20FA 00001F4E 00C0
	S <u>t</u> atics	* Parameters: void * Return: void	LPC_INT16U_LightAut 00001F52_2000
	Lige Watch	* Description: Main subroutine	00001F54 0006 if(SysInit() == 0
	Quick Watch	·	00001F56 F7FF 00001F58 FF1F
	Terminal I/O	woid main (woid)	00001F5A 2800 00001F5C D103
	Codg Coverage	ł.	
	Pro <u>f</u> iling S <u>t</u> ack	LPC_INTI6U ScreenSaver = MENU_LOW	00001F5E F002 00001F60 FA3B
Lpc213x_de	Toolhars .	fo •	C
* 🗹 Code	🗸 Status Bar		<u>^</u>
Code	@ demo.c:344.5		-
Code Code	@ demo.c:344.5		~
Debug Log	Build Breakpoints		×
Open a new R	egister window		数1 //

图 5.11 打开观察窗口

# 5.2.3 仿真调试结果

用户仿真结束,点击快捷图标 🞇 ,即可退出仿真调试状态。

用户如果选择在 RAM 中调试,程序写到芯片的 RAM 中,掉电后会丢失,再次上电目标板不会执行相应的程序。



用户如果选择在 Flash 中调试,程序写到芯片的 Flash 中,掉电后会保存,再次上电目标板会执行写入的程序。



用户如果选择带有加密配置的 Flash 中调试,程序写到芯片的 Flash 中,而且已经加密,再次上电目标板会执行写入的 程序。但是,此时不能使用任何调试工具再次进行仿真调试。除非,使用 ISP 软件全局擦除芯片,才可再次进行仿真 调试。此种方式用于用户烧写最终的程序到芯片中,而且芯片已经被加密。



# 第 6 章 TKStudio 环境仿真方法

6.1	仿真环	境设置	38
6.2	仿真调	试	39
	6.2.1	开始仿真调试	39
	6.2.2	仿真调试工具	40
	6.2.3	仿真调试结果	41

6.1 仿真环境设置

在 TKStudio 环境下打开一个编译 OK 的工程,如图 6.1 所示。点击图中 🚰 图标,进入如图 6.2 所示 的工程配置界面。

TEStudio - [main.c]		
文件(2) 編編(2) 査看(2) :	【程②】 清は② 外部设备② 工具① 智口② 帮助② -	đΧ
🔁 🖏 🤪 🖬 🖉 🕺 🖎	이 ^ ( () 승 전 전 송) - 김 왕 프 프	
🔅 🗉 🖂 🚰 shagIaFLAS	<ul> <li>Image: A state of the state of</li></ul>	Π.
0.0.01 m AA % at th	てんの間も「開」	-
1680 - + x	(ania a Starton a	×Χ
8 🚰 GPI0_Leds8		Ca
🖲 🛄 sef		0.00
8- 🛄 •. X	1 U/ Copyright (c)	^
a target	Guangzou ZLG-MCU Development Co. ,LTD.	
S S user	gredeete scheel	1
eath. c	http://www.sigmeu.com	-
	4 11 11 14	
	2 22 Bile server and a	
	R 17 Last modified Date: 2004-04-14	
	0 ** Last Version: 1.0	
	10 ** Deterintiant : The main() function example template	
	11 **	
	12 **	
	13 ** Created by: Chamingi	
	14 ** Created date: 2004-07-16	
	15 ** Version : 1.0	
	¢	>
编译室口		4 ×
Linking	(and another the second s	~
irensistion to Intel 32 bit Mex	toreat successful.	-
GPIO_LedsS. axf created - 0 erro	(s), 0 warning(s)	~
<		
🛄 编译官口 🛄 词试官口 🛄 合*	'官미   🙀 査武: 官미   🐺 査武: 官미   💽 課庭官미   💫 満用堆秋官미   🍱 代码定义官미	
Parsing successed.	LPC2132	222

图 6.1 TKStudio 主界面

在工程配置界面中,打开【Debuger】下面的 【ARM Debug】选项,选中【Hardware Emulation】, 如图 6.2 所示。



图 6.2 工程配置界面



打开【Hardware Emulation】界面,如图 6.3 所示。 点击【Add】按钮,添加 TKScope 仿真器驱动文件。

Target: DebughFLASH	General Software Situation Hardware Emulation
Competition     Constraint Constraint     Constraint Constraint     Constraint Constraint     Constraint	
< ا	8

图 6.3 硬件仿真设置窗口



系统弹出如图 6.4 所示的对话框,提示用户选择驱动 接口类型。这里选择 RDI 接口,然后点击【OK】。

Please	Select	The Dirver	Type	
۲	RDI	🔵 JDI	🔵 AGDI	
	C	ж с	Cancel	



打开 TKScope 仿真器驱动文件的安装目录(本文示例 安装路径为 D:\Keil\TKScope),选中 TKSCP\_DRV\_for\_RDI.dll,如图 6.5 所示。

查找范围(I):	🗀 TKScope	🔽 🧿 🗊 💌 🛛	
Configurat Driver ProgConfig TKSCP_DRV TKSCP_DRV TKSCP_DRV	tion g.dll ALL_for_AGDI.dll ARM_for_IAR_v4.dll ARM_for_IAR_v5.dll	TKSCP_DRV_for_RDI. dll	for_AGD
<			>
文件名(图):	TKSCP_DRV_for_RDI. dll		F @)
文件类型(T)·	Emulation Driver (* d)	1) 🗸 🗖	542

图 6.5 选择驱动文件

驱动安装完成之后,【Hardware Emulation】界面会显示当前安装的驱动选项,如图 6.6 所示。

选中 TKScope 仿真器的驱动,点击【Config】按钮, 即可进入 TKScope 仿真器设置界面,如图 6.7 所示。



无论在哪种 IDE 环境下,TKScope 仿真器设置界面都是 一样的(例如,TKStudio 环境下的图 6.7 与 Keil RealView MDK 环境下的图 3.3),设置方法也是一样的。

设置为 K9   POD-JIAG-AR	-P20   LPC2132   MXP		X
TENDER	T面/H)本 (2		
现什选择	(1) 厂商: NXP (2) 器件: IPC2132		
王要设宣	(3) 仿真器: K9 (4) POD类型: POD-JTAG-ARM-P20		
TAP设置	主要设置		
程序烧写	<ol> <li>(1) 镀仲代吗.</li> <li>(2) 缓冲数据.</li> <li>(3) 小端</li> </ol>		
初始化宏	<ul> <li>(4)使用硬件复位.</li> <li>(5)复位保持延时:50毫秒.</li> </ul>		
硬件自检	<ul> <li>(6) 复位恢复延时: 500毫秒.</li> <li>(7) 软件复位.</li> </ul>		
	(8) 目动停止。 (9) 系统时钟: 35.5000MHz.		~
	加載(保存)缺省	确认 取消	搜索

图 6.7 仿真器设置界面

#### 图 6.6 驱动安装完毕



TKScope 仿真器工作参数必须要正确设置,否则可能会导致仿真错误或失败! 具体设置方法用户请参考 3.2 小节《仿真器参数设置》,这里不再重复叙述。

# 6.2 仿真调试

仿真器工作参数设置完成之后,就可以进行仿真了。

# 6.2.1 开始仿真调试

点击【调试】菜单下的【启动/停止调试】选项,或点击快捷图标 🔍 ,即可进入仿真调试状态,如图 6.8 所示。

TEStudio - [main.c]					
三文件の 編輯の 査希の 工程	₹@) (項)	t(1) 外部设备(c)	IRD	窗口  帮助  密	_ # ×
) 🔁 🔂 🚅 🖬 🕼 🗛 🕰 👘	2 64	會口			
🔅 🖾 🖄 🥳 DebugInFLASH	٩,	启动/停止调试(Q)	Shift#F5	291002	
ି ପ୍ରାଭାଧ ରୁନ୍ଧି । 1999 - ୧୨୦ ଅନ୍ନାର - ୧୦୦		运行 (9) 忽略断点运行 (2) 停止运行	PS Ctrl+PS Shift+Ese		- × ×
e in sef e in enget e in target e in target in target	1 91 2 91 3 91 4 90 5	参選(U) 参越(y) 参出(U) 注行到当前行(U) 時時初におう	F11 F10 Shi (++F11 C+r1+F10	Copyright (o piou ILG-MCU Devel graduate school p://www.zigmcu.com	opment Co.,LTD.
	6 7 9 10 11	联点管理 插入/都除断点 算止/供能新点 禁止所有断点	A1(+P9 P9	main.c ió 1.0 The main() function e	xample template
	13 14	Annexy 19 Main 显示程序指针 ()C))	所在行	Chenning) 2004-09-16 1.0	
	<				2
编录室口					<b>+ 9 x</b>
< 	四   🛒 查!	沈宮口   🔩 査括2官	10   <b>F</b> 183	宮口   💫 调用堆枝宮口	→ 代码定义會口
启动/停止调试 (Shift+P5)		LPC2132			1,82

图 6.8 进入调试状态

进入仿真状态之后,用户可以根据实际仿真需要,选择相应的调试工具进行仿真操作。

# 表 6.1 TKStudio 运行调试工具

快捷图标	意义
Q	启动/停止调试。
5	复位。
	全速运行。
	忽略断点运行。
00	停止运行。
SI	单步。单步执行程序,进入到子程序内部。
9 <u>=</u>	布越。单步执行程序,跳过子程序,不进入到内部。
Č	步出。执行到当前子函数的结束。
<b>+</b> ()	运行到当前。程序运行到当前光标所在行时停止。
	插入/移除断点。
	启用/禁止断点。
1	禁止所有断点。
-	清除所有断点。

# 表 6.2 TKStudio 调试观察窗口工具

快捷图标	意义
<b></b>	显示/隐藏反汇编窗口。
B	观察变量窗口。
	观察存储器窗口。
<b>6</b>	观察寄存器窗口。
۵	观察 SFR 窗口。

# 6.2.3 仿真调试结果

用户仿真结束,点击【调试】菜单下的【启动/停止调试】选项,或点击快捷图标 🔍 ,即可退出仿真调试状态。

🚫 用户如果选择在 RAM 中调试,程序写到芯片的 RAM 中,掉电后会丢失,再次上电目标板不会执行相应的程序。



用户如果选择在 Flash 中调试,程序写到芯片的 Flash 中,掉电后会保存,再次上电目标板会执行写入的程序。



用户如果选择带有加密配置的 Flash 中调试,程序写到芯片的 Flash 中,而且已经加密,再次上电目标板会执行写入的 程序。但是,此时不能使用任何调试工具再次进行仿真调试。除非,使用 ISP 软件全局擦除芯片,才可再次进行仿真 调试。此种方式用于用户烧写最终的程序到芯片中,而且芯片已经被加密。

# 第7章 TKScope 仿真器技术支持



7.1	升级方法	43
7.2	联系我们	43
7.3	感谢语	44

# 7.1 升级方法

随之各个厂商新器件的推出,TKScope 仿真器会不断升级,支持新型号器件的仿真。TKScope 仿真器的升级采用的是软件升级的方 式,用户只需下载安装更新的驱动程序,即可仿真新增芯片。

TKScope 仿真器的驱动程序公布在官方网站上 <u>http://www.zlgmcu.com</u>,我们会定期维护、更新,用户可以随时、免费下载到最新 的驱动程序。

# 7.2 联系我们

感谢您选用 TKScope 仿真器,我们将竭诚为您提供技术服务,帮助您解决在使用仿真器开发过程中遇到的问题。为了尽快解决问题, 不影响您的开发进度,我们建议您通过下面的方式联系我们,获得技术支持。

电话(传真)

使用电话沟通交流是最快的解决问题的途径。请您在拨 通电话之前,整理好您的问题,将仿真器放置在计算机 和电话机旁边并处于联机状态,我们的技术支持工程师 会根据您的操作来分析您提出的问题。 技术支持电话: 020-22644360。

# 2 电子邮件

使用电子邮件您可以详细的说明描述您遇到的问题,也 可以提供测试程序或工程文件等,我们更有针对性的帮 您分析解决问题。

技术支持 Email: TKS@zlgmcu.com。

# BBS

欢迎访问周立功单片机论坛

http://www.zlgmcu.com.cn/index.asp,我们为仿真器 开设了专门的 BBS,并安排工程师轮流值日制度,对于 用户提出的全部问题,可以得到我们详尽的技术解答, 同时也可以得到其他有经验的网友朋友的回答。

# 维修服务

如果您在使用过程中遇到仿真器损坏的现象,请寄给广 州致远电子有限公司维修部,由生产厂商直接快捷的为 您服务,详细的联系方式如下: 公司:广州致远电子有限公司。 地址:广州市天河区车陂路黄洲工业区3栋2楼。 电话:020-22644245。



**注意!**由于仿真器属于特殊产品,需要用户在开发环境里正确的设置才能正常的使用,所以多数情况下仿真器出现问题 可能是设置不正确引起的,而不是仿真器损坏,用户不需要返修而解决问题。



**友情提醒**!如果您认为仿真器出现问题,我们建议您在决定返修前,与我们的技术支持工程师进行联系,确认仿真器是 否真正的出现硬件上的故障需要维修。

# 7.3 感谢语

TKScope 仿真器是广州致远电子有限公司在 2008 年推出的一款全新概念的高级专业仿真开发平台,无论是性能还是外观,都经过 工程师的精心设计,将带给用户一个全新的开发理念和感受。

TKScope 仿真器使用当今超大规模集成电路的最新技术以及规范的硬件模块化设计,全面提升 TKScope 仿真器在嵌入式系统开发中的表现能力。

TKScope 仿真器突破以往仿真器所不能,采用多种先进的仿真技术,与多种主流 IDE 环境无缝嵌接,集成众多强悍有力的功能,可 仿真芯片种类之多之广,让其不愧为仿真器界的明星产品。

再次感谢您选用 TKScope 仿真器,希望能够带给您帮助和惊喜,给您不一样的开发感觉,让您如虎添翼,项目早日成功!

如果您遇到关于嵌入式系统的任何问题,请联系我们的技术支持工程师或访问我们的主页 <u>http://www.zlgmcu.com</u>。从产品的设计 思想、元器件的选型、开发工具的使用、可靠性的设计等各个方面,我们的技术支持工程师都能为您提供全套的解决方案或良好的建议。

# 附录 TKScope 仿真器支持芯片列表

TKScope 仿真器支持众多厂商和海量芯片的仿真,而且随着新型号器件的出现不断升级,满足客户的全方面仿真需求。目前,TKScope 仿真器已经或近期陆续支持的芯片明细列表如下,厂商按照字母顺序排列,排名不分先后。

# Acer Labs

8051 内核 M6759 其它型号芯片陆续支持

#### ACTEL

Cortex-M1 内核

## Aeroflex UTMC

8051 内核 UT69RH051 其它型号芯片陆续支持

## AMD

Flash 器件 AM29F160DB, AM29F160DT, AM29F320DB, AM29F320DT, AM29F800BB, AM29F800BT, AM29LV128, AM29LV800BB, AM29LV800BT, AM29LV800DB 其它型号 Flash 器件陆续支持

#### Analog Device

- 8051 内核 ADuC812, ADuC816, ADuC824, ADuC831, ADuC832, ADuC834, ADuC836, ADuC841, ADuC842, ADuC843, ADuC845, ADuC847, ADuC848 其它型号芯片陆续支持
- ARM 内核 ADuC7019-62, ADuC7020-62, ADuC7021-62, ADuC7021-32, ADuC7022-62, ADuC7022-32, ADuC7024-62, ADuC7025-62, ADuC7025-62, ADuC7026-62, ADuC7027-62, ADuC7028-62 其它型号芯片陆续支持

## ARM

ARM 内核 ARM720T, ARM7EJ-S, ARM7TDMI, ARM7TDMI-S, ARM9TDMI, ARM920T, ARM922T, ARM926EJ-S, ARM946E-S, ARM966E-S, ARM968E-S, ARM1136, ARM1156, ARM1176, Cortex-M0, Cortex-M1, Cortex-M3 其它内核种类陆续支持

## ATMEL

8051 内核 AT80C31X2/TS80C31X2, AT80C32X2/TS80C32X2, AT80C5112, AT80C51RA2/TS80C51RA2, AT80C51RB2/TS80C51RB2, AT80C51RC2/TS80C51RC2, AT80C51RD2/TS80C51RD2, AT80C51SND1C, AT80C51SND2C, AT80C51X2/TS80C51X2, AT80C52X2/TS80C52X2, AT80C54X2/TS80C54X2, AT80C58X2/TS80C58X2, AT80SUND2CMP3B, AT83C5111, AT83C5112, AT83C5121/T83C5121, AT83C5122, AT83C5123, AT83C5127, AT83C51RA2/TS83C51RA2, AT83C51RB2/TS83C51RB2, AT83C51RC2/TS83C51RC2, AT83C51RD2/TS83C51RD2, AT83C51SND1C, AT83C51SND2C, AT83EC5122, AT83EC5123, AT83SUND2CMP3B, AT85C5121/T85C5121, AT85C5122, AT83EC5123, AT83SUND2CMP3B, AT85C5121/T85C5121, AT85C5122, AT83EC5123, AT83SUND2CMP3B, AT85C5121/T85C5121, AT85C5122, AT85C51SUND3B1, AT85C51SUND3B2, AT85C51SUND3B3, AT87C5111, AT87C5112, AT87C51RA2/TS87C51RA2, AT87C51RA2/TS87C51RA2, AT87C51RA2/TS87C51RA2, AT87C51RA2/TS87C51RA2, AT87C51RA2/TS87C51RA2, AT87C51X2/TS87C51X2, AT87C52X2/TS87C52X2, AT87C54X2/TS87C54X2, AT87C58X2/TS87C58X2, AT87F51, AT87F52, AT89C1051, AT89C2051, AT89C4051, AT89C5115/T89C5115, AT89C5121/T89C5121,

AT89C5122, AT89C5122DS, AT89C5130A-L, AT89C5130A-M, AT89C5131A-L, AT89C5131A-M, AT89C5132, AT89C51AC2/T89C51AC2, AT89C51AC3, AT89C51IC2, AT89C51ID2, AT89C51CC01/T89C51CC01, AT89C51CC02/T89C51CC02, AT89C51CC03, AT89C51RA2, AT89C51RB2, AT89C51RC2, AT89C51RC, AT89C51RD2, AT89C51RE2, AT89C51ED2, AT89C51SUND1C, AT89C51SUND2C, AT89C52, AT89C55WD, AT89SND2CMP3B, AT89LS51, AT89LS52, AT89LS53, AT89LS8252, AT89LS51, AT89LS52, AT89LS53, TS80C51U2, T83C5101, T83C5102, T83C51U2, T87C5101, T87C5102, T87C51U2, T89C51RB2, T89C51RC2, T89C51RD2 其它型号芯片陆续支持

ARM 内核 AT91SAM7A1, AT91SAM7A2, AT91SAM7A3, AT91SAM7SE32, AT91SAM7SE256, AT91SAM7SE512, AT91SAM7S16, AT91SAM7S161, AT91SAM7S32, AT91SAM7S321, AT91SAM7S64, AT91SAM7S128, AT91SAM7S256, AT91SAM7S512, AT91SAM7X128, AT91SAM7X256, AT91FR40162S, AT91FR40162SB, AT91M40800, AT91M40807, AT91R40008, AT91R40807 其它型号芯片陆续支持

#### Cortex-M3 内核

AT91SAM3U4E, AT91SAM3U2E, AT91SAM3U1E, AT91SAM3U4C, AT91SAM3U2C, AT91SAM3U1C 其它型号芯片陆续支持

- AVR 内核 ATmega16, ATmega16A, ATmega162, ATmega164P, ATmega164PA, ATmega165, ATmega165P, ATmega165PA, ATmega169, ATmega169P, ATmega169PA, ATmega32, ATmega323, ATmega324P, ATmega324PA, ATmega325, ATmega325P, ATmega3250, ATmega3250P, ATmega329P, ATmega329PA, ATmega3290, ATmega3290P, ATmega329P, ATmega329PA, ATmega3290P, ATmega3290P, ATmega3290P, ATmega3290P, ATmega3290P, ATmega645, ATmega6450, ATmega649, ATmega6440, ATmega6440, ATmega644P, ATmega644PA, ATmega645PA, ATmega6450, ATmega6450, ATmega6490, ATmega6490, ATmega128, ATmega128A, ATmega1280, ATmega1281, ATmega1284P, ATmega2560, ATmega2561, AT90CAN32, AT90CAN64, AT90CAN128, AT90USB646, AT90USB647, AT90USB1286, AT90USB1287, ATmega48, ATmega48A, ATmega88A, ATmega88A, ATmega168, ATmega168A, ATtiny13, ATtiny13A, ATtiny2313, ATtiny2313A, ATtiny24, ATtiny24A, ATtiny25, ATtiny261, ATtiny61A, ATtiny44, ATtiny44A, ATtiny45, ATtiny461, ATtiny461A, ATtiny84, ATtiny84A, ATtiny85, ATtiny861A, ATtiny861A, ATtiny67, AT90PWM29, AT90PWM216, AT90PWM38, AT90PWM316, AT90USB82, AT90USB162, ATxmega64A3, ATxmega128A1, ATxmega128A3, ATxmega192A3, ATxmega256A3, ATxmega128A3, ATxmega16A4, ATxmega16A4, ATxmega16A4, ATxmega32A4, ATxmega32D4, ATxmega64A1, ATxmega64D3, ATxmega128D3, ATxmega192D3, ATxmega256D3 其它型号芯片陆续支持
- Flash 器件 AT29C1024, AT29LV1024, AT49BV162A, AT49BV162AT, AT49BV320, AT49BV320T, AT49BV321, AT49BV321T, AT49BV1604A, AT49BV1614A, AT49BV6416, AT49LV320, AT49LV320T, AT49LV321, AT49LV321T, AT49LV1614A 其它型号 Flash 器件陆续支持

## Cirrus Logic

ARM 内核 全系列 ARM 内核,即将支持

#### CML Microcircuts

8051 内核 CMX850 其它型号芯片陆续支持

# Cybernetic Micro

8051 内核 P-51 其它型号芯片陆续支持

## Dallas

8051 内核 DS80C310, DS80C320, DS80C323, DS80C390, DS80C400, DS80C410, DS80C411, DS83C520, DS83C530, DS87C520, DS87C550, DS89C420, DS89C430, DS89C440, DS89C450 其它型号芯片陆续支持

### Faraday

ARM 内核 FA526, FA626, FIA320, FIE3360 其它型号芯片陆续支持

#### Freescale

ARM 内核 MAC7101, MAC7106, MAC7111, MAC7112, MAC7116, MAC7121, MAC7122, MAC7126, MAC7131, MAC7136, MAC7141, MAC7141, MAC7142 其它型号芯片陆续支持

## Honeywell

8051 内核 HT83C51 其它型号芯片陆续支持

#### Hynix

ARM 内核 HMS30C7202 其它型号芯片陆续支持

- GMS90xx GMS90C31, GMS90C32, GMS90C320, GMS90C51, GMS90C52, GMS90C54, GMS90C56, GMS90C58, GMS90L31, GMS90L32, GMS90L320, GMS90L51, GMS90L52, GMS90L54, GMS90L56, GMS90L58 其它型号器件陆续支持
- GMS97xx GMS97C1051, GMS97C2051, GMS97C51, GMS97C51H, GMS97C52, GMS97C52H, GMS97C54, GMS97C54H, GMS97C56, GMS97C56H, GMS97C58H, GMS97C58H, GMS97L1051, GMS97L2051, GMS97L51, GMS97L52, GMS97L54, GMS97L56, GMS97L58, GMS99C58 其它型号器件陆续支持

#### Infineon

8051 内核 C501G-1E, C501G-1R, C501G-L, C504-2E, C504-2R, C504-L, C505-2R, C505-L, C505A-2R, C505A-4R, C505A-4E, C505A-L, C505C-2R, C505C-L, C505CA-2R, C505CA-4R, C505CA-4E, C505CA-L, C505L-4E, C508-4E, C508-4R, C509-L, C511-R, C511A-R, C513-R, C513A-2R, C513A-H, C513A-R, C515-RM, C515-RN, C515-LM, C515-LN, C515A-4R, C515A-L, C515B-2R, C515C-8E, C515C-8R, C515C-L, C517A-4R, C517A-L, C540U-E, C541U-2E, SAB 80C515, SAB 80C515A, SAB 83C515A-5, SAB 80C517, SAB 80C517A, SAB 83C517A-5, SAB 80C535, SAB 80C537 其它型号芯片陆续支持

XC886 内核

XC864, XC866-1, XC866-2, XC866-4, XC878-13, XC878-16, XC886-6, XC886-8, XC888-6, XC888-8 其它型号芯片陆续支持

# C166 内核 XC166 系列

XC161CJ\_16F, XC161CS\_32F, XC164GM\_4F, XC164GM\_8F, XC164GM\_16F, XC164KM\_4F, XC164KM\_8F, XC164KM\_16F, XC164TM\_4F, XC164TM\_8F, XC164TM\_16F, XC164CM\_4F, XC164CM\_8F, XC164CM\_16F,

XC164SM\_4F, XC164SM\_8F, XC164SM\_16F, XC164LM\_4F, XC164LM\_8F, XC164LM\_16F, XC164CS\_8F, XC164CS\_16F, XC164CS\_32F, XC164D\_8F, XC164D\_16F, XC164D\_32F, XC164N\_8F, XC164N\_16F, XC164N\_32F, XC164S\_8F, XC164S\_16F, XC164S\_32F, XC167CI\_16F, XC167CI\_32F 其它型号芯片陆续支持

#### XE166 系列

XE164F\_24F, XE164F\_48F, XE164F\_72F, XE164F\_96F, XE164G\_24F, XE164G\_48F, XE164G\_72F, XE164G\_96F, XE164H\_24F, XE164H\_48F, XE164H\_72F, XE164H\_96F, XE164K\_24F, XE164K\_48F, XE164K\_72F, XE164K\_96F, XE167F\_48F, XE167F\_72F, XE167F\_96F, XE167G\_48F, XE167G\_72F, XE167G\_96F, XE167H\_48F, XE167H\_72F, XE167H\_96F, XE167K\_48F, XE167K\_72F, XE167K\_96F, XE162FM\_24F, XE162FM\_48F, XE162FM\_72F, XE162HM\_24F, XE162HM\_48F, XE162HM\_72F, XE164HM\_72F, XE164FM\_72F, XE164GM\_72F, XE164GM\_72F, XE164HM\_24F, XE164HM\_48F, XE164FM\_72F, XE164GM\_48F, XE167FM\_72F, XE164HM\_24F, XE167GM\_72F, XE167HM\_48F, XE167HM\_72F, XE167FM\_72F, XE167FM\_72F, XE167FM\_72F, XE167FM\_72F, XE167FM\_72F, XE167FM\_48F, XE167FM\_72F, XE167FM\_72F, XE167FM\_72F, XE167FM\_48F, XE167FM\_72F, XE162FN\_24F, XE162FN\_40F, XE162HN\_16F, XE162HN\_24F, XE164FN\_24F, XE164FN\_40F, XE162HN\_16F, XE164HN\_24F, XE164HN\_40F, XE164HN\_16F, XE164HN\_40F, XE164HN\_24F, XE164HN\_24F, XE164KN\_40F 其它型号芯片陆续支持

#### XC2000 系列

XC2238M\_56F,XC2239M\_72F,XC2264\_56F,XC2264\_72F,XC2264\_96F,XC2267\_56F,XC2267\_72F,XC2267\_96F, XC2263M\_56F,XC2263M\_72F,XC2263M\_104F,XC2264M\_56F,XC2264M\_72F,XC2264M\_104F,XC2265M\_56F, XC2265M\_72F,XC2265M\_104F,XC2267M\_56F,XC2267M\_72F,XC2267M\_104F,XC2268M\_56F,XC2268M\_72F, XC2268M\_104F,XC2285\_56F,XC2285\_72F,XC2285\_96F,XC2286\_56F,XC2286\_72F,XC2286\_96F,XC2287\_56F, XC2287\_72F,XC2287\_96F,XC2285M\_56F,XC2285M\_72F,XC2285M104F,XC2286M\_56F,XC2286M\_72F, XC2286M\_104F,XC2287M\_56F,XC2285M\_56F,XC2285M\_72F,XC2285M104F,XC2286M\_56F,XC2286M\_72F, XC2286M\_104F,XC2287M\_56F,XC2287M\_72F,XC2287M\_104F,XC2336A\_56F,XC2336A\_72F,XC2365\_48F, XC2365\_56F,XC2365\_72F,XC2361A\_56F,XC2361A\_72F,XC2363A\_56F,XC2363A\_72F,XC2364A\_56F, XC2364A\_72F,XC2364A\_104F,XC2365A\_56F,XC2365A\_72F,XC2365A\_104F,XC2387\_72F,XC2385A\_56F, XC2385A\_72F,XC2385A\_104F,XC2387A\_56F,XC2387A\_72F,XC2387A\_104F,XC2735\_40F,XC2735\_72F, XC2765\_72F,XC2765\_104F,XC2766\_96F,XC2785\_72F,XC2785\_104F,XC2786\_96F 其它型号芯片陆续支持

## ISSI

8051 内核 IS80C31, IS80C32, IS80C51, IS80C52, IS80LV31, IS80LV32, IS80LV51, IS80LV52, IS89C51, IS89C52 其它型号芯片陆续支持

#### Luminary Micro

Cortex-M3 内核

LM3S101, LM3S102 , LM3S300, LM3S301, LM3S308, LM3S310, LM3S315, LM3S316, LM3S317, LM3S328, LM3S600, LM3S601, LM3S608, LM3S610, LM3S611, LM3S612, LM3S613, LM3S615, LM3S617, LM3S618, LM3S628, LM3S800, LM3S801, LM3S808, LM3S811, LM3S812, LM3S815, LM3S817, LM3S818, LM3S828, LM3S1110, LM3S1133, LM3S1138, LM3S1150, LM3S1162, LM3S1165, LM3S132, LM3S1435, LM3S1439, LM3S1512, LM3S1538, LM3S1601, LM3S1607, LM3S1608, LM3S1620, LM3S1625, LM3S1626, LM3S1627, LM3S1635, LM3S1637, LM3S1751, LM3S1776, LM3S1608, LM3S1911, LM3S1918, LM3S1937, LM3S1958, LM3S1960, LM3S1968, LM3S2016, LM3S2110, LM3S2139, LM3S2276, LM3S2410, LM3S2412, LM3S2432, LM3S2533, LM3S2601, LM3S2608, LM3S2616, LM3S2620, LM3S2637, LM3S2651, LM3S2671, LM3S2678, LM3S2730, LM3S2739, LM3S2776, LM3S2611, LM3S3759, LM3S2768, LM3S5732, LM3S5737, LM3S5747, LM3S5749, LM3S5652, LM3S562, LM3S5752, LM3S5757, LM3S5762, LM3S5767, LM3S5768, LM3S5769, LM3S6100, LM3S6110, LM3S6420, LM3S6422, LM3S6432, LM3S6637, LM3S6611, LM3S6613, LM3S6633, LM3S6637, LM3S6730, LM3S6753, LM3S6816, LM3S6911, LM3S6614, LM3S6614, LM3S6633, LM3S6637, LM3S6730, LM3S6753, LM3S6911, LM3S66916, LM3S6637, LM3S6637, LM3S6753, LM3S6816, LM3S6911, LM3S66916, LM3S6637, LM3S6637, LM3S6753, LM3S6816, LM3S6911, LM3S6916, LM3S6633, LM3S6637, LM3S6730, LM3S6753, LM3S6816, LM3S6911, LM3S6916, LM3S6637, LM3S6637, LM3S6753, LM3S6631, LM3S6637, LM3S6753, LM3S6637, LM3S6637, LM3S6753, LM3S6631, LM3S6637, LM3S6753, LM3S6637, LM3S6753, LM3S6631, LM3S6637, LM3S6637, LM3S6753, LM3S6637, LM3S6637, LM3S6753, LM3S6637, LM3S6637, LM3S6753, LM3S6637, LM3S6637, LM3S6753, LM3S6637, LM3S6753, LM3S6637, LM3S6753, LM3S6637, LM3S6753, LM3S6753, LM3S6637, LM3S6

LM3S6918, LM3S6938, LM3S6950, LM3S6952, LM3S6965, LM3S8530, LM3S8538, LM3S8630, LM3S8730, LM3S8733, LM3S8738, LM3S8930, LM3S8933, LM3S8938, LM3S8962, LM3S8970, LM3S8971, LM3S9790, LM3S9792, LM3S9997, LM3S9B90, LM3S9B92, LM3S9B95, LM3S9B96, LM3S9L97 其它型号芯片陆续支持

## MARVELL (Intel)

8051 内核 8031AH, 8032AH, 8051AH, 8052AH, 80C151SB, 80C152JA, 80C152JB, 80C152JC, 80C152JD, 80C31BH, 80C32, 80C51BH, 80C51FA, 80C51GB, 80C51RA, 80C51SL\_AH, 80C51SL\_AL, 80C52, 80C54, 80C58, 80L32, 80L51FA, 80L52, 80L54, 80L58, 81C51SL\_AH, 81C51SL\_AL, 83C151SA, 83C151SB, 83C152JA, 83C152JB, 83C152JD, 83C51FA, 83C51FB, 83C51FC, 83C51GB, 83C51RA, 83C51RB, 83C51RC, 83C51SL\_AH, 83C51SL\_AL, 83L51FA, 83L51FB, 83L51FC, 87C151SA, 87C151SB, 87C51BH, 87C51FA, 873L51FB, 87C51RA, 87C51RB, 87C51RA, 87C51RA, 87C51RA, 87C51RA, 87L51FA, 87L51FA, 87L51FC, 87C51SL\_AH, 87C51SL\_AL, 87C52, 87C54, 87C58, 87L51FA, 873L51FB, 87L51FC, 87L52, 87L54, 87L58 其它型号芯片陆续支持

#### XSCALE 内核

PXA255,PXA270 其它型号芯片陆续支持

# Megawin

8051 内核 MPC89E515A, MPC89E51A, MPC89E52A, MPC89E53A, MPC89E54A, MPC89E58A, MPC89L515A, MPC89L516X2, MPC89L51A, MPC89L52A, MPC89L53A, MPC89L54A, MPC89L556X2, MPC89L58A 其它型号芯片陆续支持

## NUVOTON (WinBOND)

8051 内核 W77C32/W77C032, W77E58/W77E058, W77E516, W77E532, W77LE58/W77L058, W77LE516/W77L516, W77LE532/W77L532, W78C32/W78C032, W78C51/W78C051, W78C521/W78C052, W78C54/W78C054, W78C801, W78E51/W78E051, W78E52/W78E052, W78E54/W78E054, W78E58/W78E058, W78E365, W78E516, W78E858, W78ERD2, W78IRD2, W78L32, W78L51, W78L52, W78L54, W78L801, W78LE51/W78L051, W78LE52/W78L052, W78LE54/W78L054, W78LE58/W78L058, W78LE516/W78L516, W78LE812/L812, W79E201, W79E532, W79E548, W79E549, W79E632, W79E633, W79E648, W79E649, W79L532, W79L548, W79L549, W79L632, W79L633, W79L648, W79L649 其它型号芯片陆续支持

#### ARM 内核 NUC501

其它型号芯片陆续支持

#### Cortex-M0 内核

NUC100LE3AN, NUC100RE3AN, NUC100RD3AN, NUC100VE3AN, NUC100VD3AN, NUC100VD2AN, NUC100LD2AN, NUC100LD1AN, NUC100LC1AN, NUC100RD2AN, NUC100RD1AN, NUC100RC1AN, NUC120LE3AN, NUC120LD3AN, NUC120RE3AN, NUC120RD3AN, NUC120VE3AN, NUC120VD3AN, NUC120VD2AN, NUC120LD1AN, NUC120LC1AN, NUC120RD2AN, NUC120RD1AN, NUC120RC1AN, NUC130LE3AN, NUC130LD3AN, NUC130LD2AN, NUC130RE3AN, NUC130RD3AN, NUC130RD2AN, NUC130VE3AN, NUC130VD3AN, NUC130VD2AN, NUC140LE3AN, NUC140LD3AN, NUC140RD3AN, NUC140RD3AN, NUC140VD3AN, NUC1

Flash 器件 W19B320AB, W19B320AT, W19B320BB, W19B320BT 其它型号 Flash 器件陆续支持

#### NXP

- 8051 内核 P80C31, P80C31X2, P80C32, P80C32X2, P80C51FA, P80C51RA+, P80C552, P80C554, P80C557E4, P80C557E6, P80C557E8, P80C591, P80C592, P80CE598, P80C654X2, P80C660X2, P80C661X2, P87C51FA, P87C51FB, P87C51FB, P87C51MB2, P87C51MC2, P87C51MB2-02, P87C51MC2-02, P87C51RA+, P87C51RA+, P87C51RC+, P87C51RD+, P87C51RA2, P87C51RB2, P87C51RC2, P87C51RD2, P87C51, P87C51X2, P87C52, P87C52X2, P87C54, P87C54X2, P87C58X2, P87C552, P87C554, P87C557E6, P87C557E8, P87C591, P87C592, P87C598, P87C654X2, P87C660X2, P87C661X2, P89C51, P89C51X2, P89C52, P89C52X2, P89C54, P89C54X2, P89C58, P89C58X2, P89C51RA2, P89C51RD2, P89C51RC2, P89C51RD2, P89C51RA2H, P89C51RB2H, P89C51RC2H, 89C51RD2H, P89C51RC2+, P89C51RD2+, P89LV51RD2, P89C60X2, P89C61X2, P89C557E4, P89C660, P89C662, P89C664, P89C668, P89C669, P89V52X2, P89V660, P89V662, P89V664 其它型号芯片陆续支持
- ARM 内核 LPC2101, LPC2102, LPC2103, LPC2104, LPC2105, LPC2106, LPC2109, LPC2114, LPC2119, LPC2124, LPC2129, LPC2194, LPC2131, LPC2132, LPC2134, LPC2136, LPC2138, LPC2141, LPC2142, LPC2144, LPC2146, LPC2148, LPC2157, LPC2158, LPC2194, LPC2210, LPC2212, LPC2214, LPC2220, LPC2290, LPC2292, LPC2294, LPC2364, LPC2365, LPC2366, LPC2367, LPC2368, LPC2377, LPC2378, LPC2387, LPC2388, LPC2458, LPC2460, LPC2468, LPC2470, LPC2478, LPC2880, LPC2888, LPC2917, LPC2919, LPC3130, LPC3131, LPC3141, LPC3143, LPC3152, LPC3154, LPC3180, LPC3220, LPC3230, LPC3240, LPC3250, LH7A400, LH7A404 其它型号芯片陆续支持

#### Cortex-M3 内核

LPC1751, LPC1752, LPC1754, LPC1756, LPC1758, LPC1764, LPC1765, LPC1766, LPC1768, LPC1311, LPC1313, LPC1342, LPC1343 其它型号芯片陆续支持

#### Cortex-M0 内核

LPC1111x101, LPC1111x201, LPC1112x101, LPC1112x201, LPC1113x101, LPC1113x201, LPC1114x101, LPC1114x201 其它型号芯片陆续支持

## OKI

8051 内核 MSM80C154S, MSM80C31F, MSM80C51F, MSM83C154S 其它型号芯片陆续支持

ARM 内核 全系列 ARM 内核,即将支持

#### Samsung

ARM 内核 S3C44B0X, S3C2410A, S3C2416X, S3C2440A, S3C4510, S3C6410 其它型号芯片陆续支持

### Silicon Labs

#### 8051 内核 C8051F00x/01x 系列

C8051F000, C8051F001, C8051F002, C8051F005, C8051F006, C8051F007, C8051F010, C8051F011, C8051F012, C8051F015, C8051F016, C8051F017 其它型号芯片陆续支持

C8051F02x 系列 C8051F020, C8051F021, C8051F022, C8051F023 其它型号芯片陆续支持 C8051F04x 系列 C8051F040, C8051F041, C8051F042, C8051F043, C8051F044, C8051F045, C8051F046, C8051F047 其它型号芯片陆续支持 C8051F06x 系列 C8051F060, C8051F061, C8051F062, C8051F063, C8051F064, C8051F065, C8051F066, C8051F067 其它型号芯片陆续支持 C8051F2xx 系列 C8051F206, C8051F220, C8051F221, C8051F226, C8051F230, C8051F231, C8051F236 其它型号芯片陆续支持 C8051F30x 系列 C8051F300, C8051F301, C8051F302, C8051F303, C8051F304, C8051F305 其它型号芯片陆续支持 C8051F31x 系列 C8051F310, C8051F311, C8051F312, C8051F313, C8051F314, C8051F315, C8051F316, C8051F317 其它型号芯片陆续支持 C8051F32x 系列 C8051F320, C8051F321, C8051F326, C8051F327 其它型号芯片陆续支持 C8051F33x 系列 C8051F330, C8051F331, C8051F332, C8051F333, C8051F334, C8051F335, C8051F336, C8051F337, C8051F338, C8051F339 其它型号芯片陆续支持 C8051F34x 系列 C8051F340, C8051F341, C8051F342, C8051F343, C8051F344, C8051F345, C8051F346, C8051F347, C8051F348, C8051F349, C8051F34A, C8051F34B 其它型号芯片陆续支持 C8051F35x 系列 C8051F350, C8051F351, C8051F352, C8051F353 其它型号芯片陆续支持 C8051F36x 系列 C8051F360, C8051F361, C8051F362, C8051F363, C8051F364, C8051F365, C8051F366, C8051F367, C8051F368, C8051F369 其它型号芯片陆续支持 C8051F41x 系列 C8051F410, C8051F411, C8051F412, C8051F413 其它型号芯片陆续支持 C8051F50x/51x 系列 C8051F500, C8051F501, C8051F502, C8051F503, C8051F504, C8051F505, C8051F506, C8051F507, C8051F508, C8051F509, C8051F510, C8051F511 其它型号芯片陆续支持 C8051F52x/53x 系列 C8051F520, C8051F521, C8051F523, C8051F524, C8051F526, C8051F527, C8051F530, C8051F531, C8051F533, C8051F534, C8051F536, C8051F537 其它型号芯片陆续支持 C8051F70x/71x 系列

C8051F700, C8051F701, C8051F702, C8051F703, C8051F704, C8051F705, C8051F706, C8051F707, C8051F708, C8051F709, C8051F710, C8051F711, C8051F712, C8051F713, C8051F714, C8051F715 其它型号芯片陆续支持 C8051F920, C8051F921, C8051F930, C8051F931 其它型号芯片陆续支持 C8051F12x/13x 系列 C8051F120, C8051F121, C8051F122, C8051F123, C8051F124, C8051F125, C8051F126, C8051F127, C8051F130, C8051F131, C8051F132, C8051F133 其它型号芯片陆续支持 C8051F580, C8051F581, C8051F582, C8051F583, C8051F584, C8051F585, C8051F586, C8051F587, C8051F588, C8051F589, C8051F590, C8051F591 其它型号芯片陆续支持

# SST

- 8051 内核 SST89E516RD, SST89E516RD2, SST89E51RC, SST89E52RC, SST89E52RD, SST89E52RD2, SST89E54RC, SST89E54RD, SST89E54RD2, SST89E54RD2, SST89E54RD4, SST89E58RD4, SST89E58RD4, SST89E554RC, SST89E564RD, SST89V516RD, SST89V516RD2, SST89V51RC, SST89V52RD, SST89V52RD2, SST89V52RD2, SST89V54RD4, SST89V54RD4, SST89V54RD4, SST89V58RD4, SST89V58RD4, SST89V58RD4, SST89V58RD4, SST89V58RD4, SST89V58RD4, SST89V58RD4, SST89V554RC, SST89V554RC, SST89V564RD 其它型号芯片陆续支持
- Flash 器件 SST36VF1601, SST36VF1602, SST36VF3203, SST36VF3204, SST39LF200A, SST39LF400A, SST39LF800A, SST39VF1601, SST39VF1602, SST39VF3201, SST39VF3202, SST39VF6401, SST39VF6402, SST39WF400A, SST39WF800A, SST39WF1601, SST39WF1602 其它型号 Flash 器件陆续支持

#### ST

ARM 内核 STR710FZ1, STR710FZ2, STR711FR0, STR711FR1, STR711FR2, STR712FR0, STR712FR1, STR712FR2, STR715FR0, STR730FZ1, STR730FZ2, STR731FV0, STR731FV1, STR731FV2, STR735FZ1, STR735FZ2, STR736FV0, STR736FV1, STR736FV2, STR750FV0, STR750FV1, STR750FV2, STR751FR0, STR751FR1, STR751FR2, STR752FR0, STR752FR1, STR752FR2, STR755FR0, STR755FR1, STR755FR2, STR755FV0, STR755FV1, STR755FV2, STR910FM32X6, STR910FW32X6, STR910FAZ32H6, STR911FM42X6, STR911FM44X6, STR912FW42X6, STR912FW44X6, STR912FAZ42H6 其它型号芯片陆续支持

#### Cortex-M3 内核

STM32F101C4, STM32F101R4, STM32F101T4, STM32F101C6, STM32F101R6, STM32F101T6, STM32F101C8, STM32F101R8, STM32F101V8, STM32F101T8, STM32F101RB, STM32F101VB, STM32F101CB, STM32F101RC, STM32F101VC, STM32F101ZC, STM32F101RD, STM32F101VD, STM32F101ZD, STM32F101RE, STM32F101VE, STM32F101ZE, STM32F102C4, STM32F102R4, STM32F102C6, STM32F102R6, STM32F102C8, STM32F102R8, STM32F102CB, STM32F102RB, STM32F103C4, STM32F103R4, STM32F103T4, STM32F103C6, STM32F103R6, STM32F103T6, STM32F103C8, STM32F103R8, STM32F103V8, STM32F103T8, STM32F103RB, STM32F103VB, STM32F103CB, STM32F103RC, STM32F103VC, STM32F103ZC, STM32F103RD, STM32F103VD, STM32F103ZD, STM32F103RE, STM32F103VE, STM32F103ZE 其它型号芯片陆续支持

# STC

8051 内核 STC89C51RC, STC89C52RC, STC89C53RC, STC89C54RD+, STC89C55RD+, STC89C58RD+, STC89C516RD+, STC89LE51AD, STC89LE52AD, STC89LE54AD, STC89LE58AD, STC89LE516AD, STC89LE51RC, STC89LE52RC, STC89LE54RD+, STC89LE58RD+, STC89LE516RD+, STC89LE516X2 其它型号芯片陆续支持

## SyncMOS

8051 内核 SM5964AL, SM5964C, SM59128C, SM59264, SM7964AL, SM7964C, SM79108C, SM79108L, SM79164C, SM79164L, SM79164V, SM80C51C, SM80C51L, SM80C52C, SM80C52L, SM80C58C, SM80C58L, SM8951AC, SM8951AL, SM8951BC, SM8951BL, SM8952AC, SM8952AL, SM8954AC, SM8954AL, SM8958AC, SM8958AL, SM89516AC, SM89516AL, SM89516C, SM89516L, SM894051L, SM89508R1C, SM89S08R1L, SM89S16R1C, SM89S16R1L, SM89S16R1L, SM89T08R1C, SM89T08R1L 其它型号芯片陆续支持

## TI

- 8051 内核 MSC1200Y2, MSC1200Y3, MSC1201Y2, MSC1201Y3, MSC1202Y2, MSC1202Y3, MSC1210Y2, MSC1210Y3, MSC1210Y4, MSC1210Y5, MSC1211Y2, MSC1211Y3, MSC1211Y4, MSC1211Y5, MSC1212Y2, MSC1212Y3, MSC1212Y4, MSC1212Y5, MSC1213Y2, MSC1213Y3, MSC1213Y4, MSC1213Y5, MSC1214Y2, MSC1214Y3, MSC1214Y4, MSC1214Y5 其它型号芯片陆续支持
- ARM 内核 TMS470R1A64, TMS470R1A128, TMS470R1A256 其它型号芯片陆续支持

# DSP 内核 DaVinci<sup>™</sup>数字媒体处理器

TMS320DM6467-594, TMS320DM6467-729, TMS320DM6446-594, TMS320DM6446-513, TMS320DM6443-594, TMS320DM6441-513, TMS320DM6441-405, TMS320DM6437-700, TMS320DM6437-600, TMS320DM6437-500, TMS320DM6437-400, TMS320DM6435-700, TMS320DM6435-600, TMS320DM6435-500, TMS320DM6435-400, TMS320DM6433-700, TMS320DM6433-600, TMS320DM6433-500, TMS320DM6433-400, TMS320DM6433-700, TMS320DM6433-600, TMS320DM6433-500, TMS320DM6433-400, TMS320DM6443-720, TMS320DM6447-900, TMS320DM6447-720, TMS320DM642-720, TMS320DM642-600, TMS320DM642-500, TMS320DM641-600, TMS320DM641-500, TMS320DM640-400, TMS320DM642-500, TMS320DM355-216, TMS320DM355-135, TMS320DM335-216, TMS320DM335-135

## OMAP<sup>™</sup>应用处理器

OMAP3530, OMAP3525, OMAP3515, OMAP3503, OMAP-L137

## C2000<sup>™</sup> 系列 32 位实时 MCU

TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28035, TMS320F28027, TMS320F28023, TMS320C2801, TMS320C2802, TMS320C2810, TMS320C2811, TMS320C2812, TMS320F2801-100, TMS320F2801-60, TMS320F28015, TMS320F28016, TMS320F2802-100, TMS320LF2407A, TMS320LF2406A, TMS320LF2403A, TMS320LF2402A, TMS320LF2401A, TMS320LC2406A, TMS320LC2404A, TMS320LC2403A, TMS320LC2402A, TMS320LC2401A

## C5000<sup>™</sup>低功耗 DSP

TMS320VC5510A-200, TMS320VC5510A-160, TMS320VC5509A-200, TMS320VC5507-200, TMS320VC5506-108, TMS320VC5503-200, TMS320VC5502-300, TMS320VC5502-200, TMS320VC5501-300, TMS320VC549-120, TMS320VC549-100, TMS320VC5471, TMS320VC5470, TMS320VC5441-532, TMS320VC5421-200, TMS320VC5416-160, TMS320VC5416-120, TMS320VC5410A-160, TMS320VC5410A-120, TMS320VC5410-100, TMS320VC5409A-160, TMS320VC5409A-120, TMS320VC5409-80, TMS320VC5409-100, TMS320VC5407-120, TMS320VC5404-120, TMS320VC5402A-160, TMS320VC5402-100

# C6000<sup>™</sup> DSP 平台

TMS320C6474 , TM	1S320C6455-850 , TI	VIS320C6455-720 ,	TMS320C6455-1200 ,	TMS320C6455-1000 ,
TMS320C6454-850,	TMS320C6454-720,	TMS320C6454-1000	, TMS320C6452-900 ,	TMS320C6452-720,
TMS320C6416T-850,	TMS320C6416T-720,	TMS320C6416T-600	, TMS320C6416T-1000	, TMS320C6416-7E3,
TMS320C6416-6E3,	TMS320C6416-5E0,	TMS320C6415T-850,	TMS320C6415T-720,	TMS320C6415T-600,
TMS320C6415T-1000	, TMS320C6415-7E3	, TMS320C6415-6E3	, TMS320C6415-5E0,	TMS320C6414T-850,
TMS320C6414T-720,	TMS320C6414T-600,	TMS320C6414T-1000,	TMS320C6414-7E3, T	MS320C6414-6E3
C6000 <sup>™</sup> 性能值 DSP				
TMS320C6424-700 ,	TMS320C6424-600 ,	TMS320C6424-500	, TMS320C6424-400 ,	TMS320C6421-700 ,
TMS320C6421-600 ,	TMS320C6421-500 ,	TMS320C6421-400	, TMS320C6418-600 ,	TMS320C6418-500,
TMS320C6413-500,	TMS320C6412-720 ,	TMS320C6412-600	, TMS320C6412-500 ,	TMS320C6411-300,
TMS320C6410-400,	TMS320C6211B-167,	TMS320C6211B-150	, TMS320C6205-200,	TMS320C6204-200,
TMS320C6203B-300,	TMS320C6203B-250,	TMS320C6202B-300,	TMS320C6202B-250, 1	FMS320C6201-200
C6000 <sup>™</sup> 浮点 DSP				
TMS320C6745-200,	TMS320C6745-300,	TMS320C6747-200,	TMS320C6747-300,	TMS320C6727B-350,
TMS320C6727B-300,	TMS320C6727B-275,	TMS320C6727B-250	, TMS320C6726B-266,	TMS320C6726B-225,
TMS320C6722B-250,	TMS320C6722B-225,	TMS320C6722B-200	), TMS320C6720-200,	TMS320C6713B-300,
TMS320C6713B-225,	TMS320C6713B-200,	TMS320C6713B-167	, TMS320C6712D-150,	TMS320C6711D-250,
TMS320C6711D-200,	TMS320C6711D-167,	TMS320C6701-167,	TMS320C6701-150, SM3	320C6713B-EP
其它型号芯片陆续支持				

# TOSHIBA

Cortex-M3 内核

TMPM330FDFG, TMPM330FYFG, TMPM330FWFG, TMPM332FWUG, TMPM370FYDFG, TMPM370FYFG, TMPM380FYDFG, TMPM380FYFG 其它型号芯片陆续支持