

# AK100 高性能 ARM 专用仿真器

## SuperARM ICE

AK100 Emulator



AK100 傲世群雄，ARM 仿真器领导品牌

## User Guide

---

用户指南

## 产品制造商与技术支持

---

### 广州致远电子股份有限公司

地址：广州市天河区车陂路黄洲工业区 3 栋 2 楼

邮编：510660

电话：(020) 22644372 28872453 (销售服务)

(020) 22644360 28267808 (技术支持)

传真：(020) 38601859

E-mail: [TKS@zlgmcu.com](mailto:TKS@zlgmcu.com) (技术支持)

网址: [www.embedtools.com](http://www.embedtools.com)

## 销售与服务网络

---

### 广州周立功单片机科技有限公司

地址：广州市天河区天河北路 689 号光大银行大厦 12 楼 F4 室

邮编：510630

电话：(020) 38730916 38730917 38730976 38730977

传真：(020) 38730925

网址: [www.zlgmcu.com](http://www.zlgmcu.com)

### 广州专卖店

地址：广州市天河区新赛格电子城 203~204 室

电话：(020) 87578634 87569917 87578842

传真：(020) 87578842

### 深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室

电话：(0755) 83781768 83781788 83782922

传真：(0755) 83793285

### 成都周立功

地址：成都市一环路南二段 1 号数码同人港 401 室

(磨子桥立交西北角)

电话：(028) 85439836 85432683 85437446

传真：(028) 85437896

### 重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦 1611 室

(赛格电子市场)

电话：(023) 68796438 68796439

传真：(023) 68796439

### 武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室

(华中电脑数码市场)

电话：(027) 87168497 87168297 87168397

传真：(027) 87163755

### 北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座 1207~1208 室

(中发电子市场斜对面)

电话：(010) 62536178 62536179 82628073

传真：(010) 82614433

### 上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室

电话：(021) 53083452 53083453 53083496

传真：(021) 53083491

### 杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571) 28139611 28139612 28139613

传真：(0571) 28139621

### 南京周立功

地址：南京市珠江路 280 号珠江大厦 2006 室

电话：(025) 83613221 83603005 83603500

传真：(025) 83613271

### 西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室

电话：(029) 87881296 83063000 87881295

传真：(029) 87880865

# 目录

## 第 1 章 认识 AK100 仿真器

1.1 功能简介	5
1.2 性能特点	6
1.3 同类比较	7

## 第 2 章 使用 AK100 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 仿真环境设置	15
3.2 仿真器参数设置	16
3.2.1 硬件选择	16
3.2.2 主要设置	17
3.2.3 TAP 设置	18
3.2.4 程序烧写	19
3.2.5 初始化宏	22
3.2.6 硬件自检	24
3.3 仿真调试	24
3.3.1 开始仿真调试	24
3.3.2 仿真调试工具	25
3.3.3 仿真调试结果	26

## 第 4 章 ADS 环境仿真方法

4.1 添加驱动文件	28
4.2 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

## 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

## 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

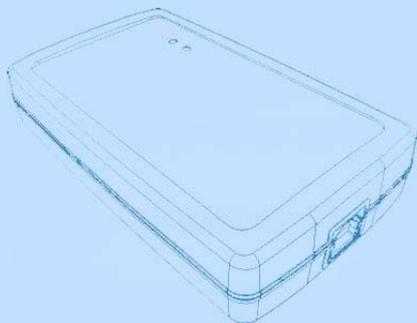
## 第 7 章 AK100 仿真器技术支持

7.1 升级方法	43
7.2 联系我们	43
7.3 感谢语	44

## 附录 AK100 仿真器支持芯片列表

# 第 1 章

## 认识 AK100 仿真器



1.1 功能简介	5
1.2 性能特点	6
1.3 同类比较	7

## 1.1 功能简介

AK100 仿真器是广州致远电子有限公司 2009 年隆重推出的一款高性能 ARM 专用仿真器，支持 ARM7、ARM9、Cortex-M0、Cortex-M1、Cortex-M3、XSCALE 等内核的全系列仿真，包括 Thumb 模式。随之不断升级，后续还会支持 ARM10、ARM11、Cortex-R4、Cortex-A8 等内核的全系列仿真。



目前，AK100 仿真器支持的内核种类如下：

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

AK100 仿真器与多种主流 IDE 环境无缝嵌接，支持 Keil、ADS、IAR、TKStudio 等，并具备其高级调试功能，保证您的开发平台始终如一。



目前，AK100 仿真器支持的 IDE 环境如下：

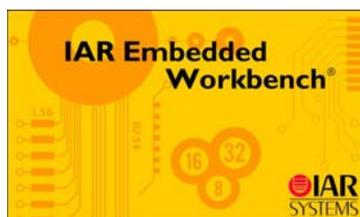
- **Keil,** Keil 公司，英文界面，8051/251/C166/ARM 编译/调试环境。
- **ADS,** ARM 公司，英文界面，全 ARM 内核编译/调试环境。
- **IAR,** IAR 公司，英文界面，多内核编译/调试环境。
- **TKStudio,** 致远公司，中/英文界面，多内核编译/调试环境，强大内置编辑器。



Keil RealView MDK



ARM ADS



IAR EWARM



TKStudio

AK100 仿真器采用 **USB2.0**（High Speed）高速通讯接口，快速下载用户程序代码。同时，PC 机 USB 口取电，省去沉重的电源适配器，外形美观小巧，携带方便。



AK100 正面图

## 1.2 性能特点

AK100 仿真器是 TKScope SuperARM ICE AK 系列的仿真产品，只专注于 ARM 内核的仿真。其性能与 TKScope 嵌入式综合仿真开发平台仿真 ARM 内核性能是一样的，可以说是 TKScope 仿真器的 ARM 单机版。



### TKScope 简介!

TKScope 嵌入式综合仿真开发平台，支持 ARM、DSP、AVR、8051、C166、C8051F、MX 等内核全系列仿真，是一款全新概念的高级专业通用型仿真开发平台；与多种主流 IDE 环境无缝嵌接，支持 TKStudio、zlglogic、Keil、ADS、IAR、RealView、AVR、CCS 等；同时，内嵌专业的逻辑分析仪，zlglogic 高级软件全面支持。

### AK100 仿真器硬件指标

- USB2.0 (High Speed) 高速通讯接口。
- 下载速度达到 600KB/S。
- 标准 Windows USB 驱动支持，即插即用。
- 标准 20-pin JTAG 接口与目标板连接，支持热插拔。
- PC 机 USB 口取电，省去沉重的电源适配器，小巧便携。
- 人性化的指示灯提示当前工作状态。
- 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。



AK100 仿真器

### AK100 仿真器功能特性

- 支持全系列 ARM 内核仿真，ARM7/ARM9/Cortex-M0/Cortex-M1/Cortex-M3/XSCALE 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，Keil/ADS/IAR/TKStudio/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。

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

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 *	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 *	NOR, NAND, SPI	NOR, NAND, SPI	NOR, NAND, SPI	NOR
用户添加 Flash 算法 *	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 *	<=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

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。

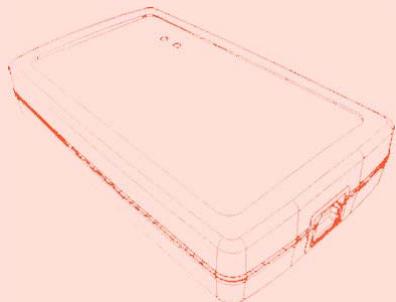
### TKScope K9/AK100/Ulink2/J-link 下载速度测试结果 (LPC2138)

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 AK100 仿真器



2.1 安装驱动

10

2.2 硬件连接

11

## 2.1 安装驱动

AK100 仿真器使用之前必须安装驱动程序，否则，无法正常工作！AK100 仿真器的驱动程序与 TKScope 系列仿真器的驱动程序是一样的。



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。

1 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。

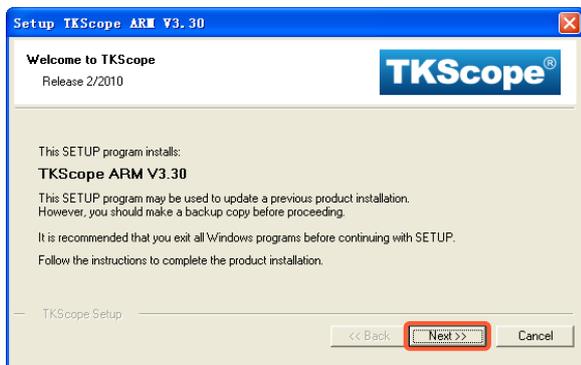


图 2.1 安装驱动提示框

2 本文示例 Keil RealView MDK 开发环境安装在 D 盘，所以，AK100 仿真器的驱动安装路径为 D:\Keil，如图 2.2 所示。

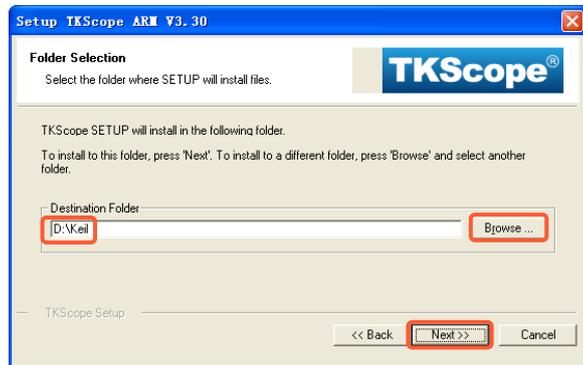


图 2.2 指定驱动安装路径

3 AK100 仿真器驱动安装完毕，建议用户安装来自微软的 vc8 的实时运行库。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。

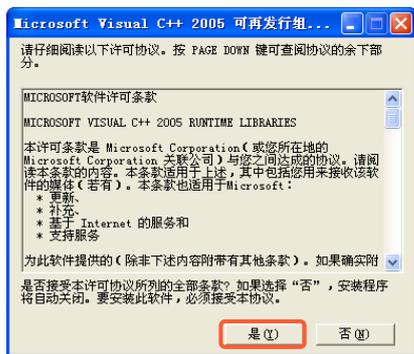


图 2.3 安装实时运行库

4 至此，AK100 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。

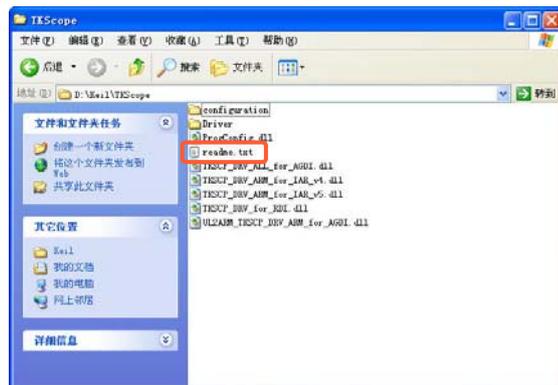


图 2.4 驱动安装目录文件



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 Uvsvion4/Uvsvion3/Uvsvion2 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 AK100 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

AK100 仿真器通过 USB 接口与计算机连接，即插即用。



USB 连接线接通后，AK100 仿真器 Power 指示灯点亮，Busy 指示灯点亮一下即熄灭（通讯过程中 Busy 指示灯会不断的闪烁）。

### 连接仿真器和目标板

AK100 仿真器通过 20Pin 扁平电缆与目标板 JTAG 接口连接，支持热插拔。



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

AK100 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

2

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。



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 AK100 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\AK100 Driver\WinXP），然后点击【确定】。



图 2.7 指定驱动具体位置

4

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。

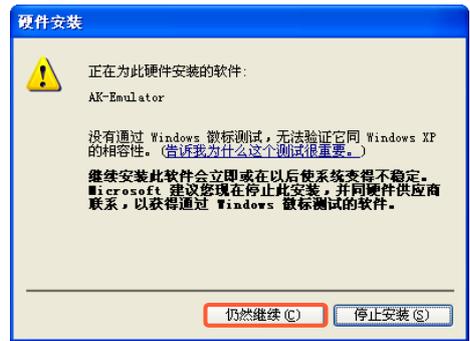


图 2.8 硬件安装过程

5

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

1

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。

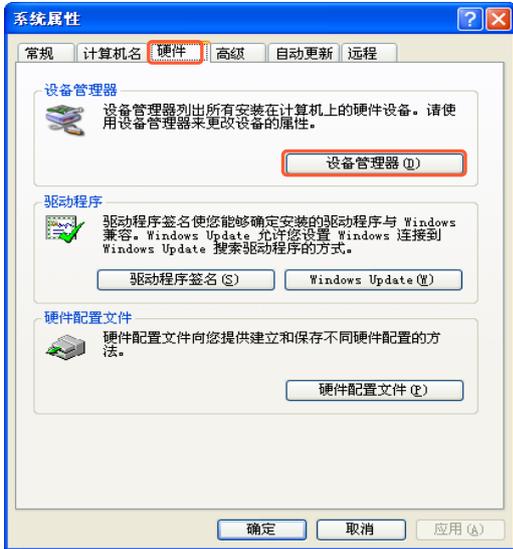


图 2.10 系统属性

3

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。

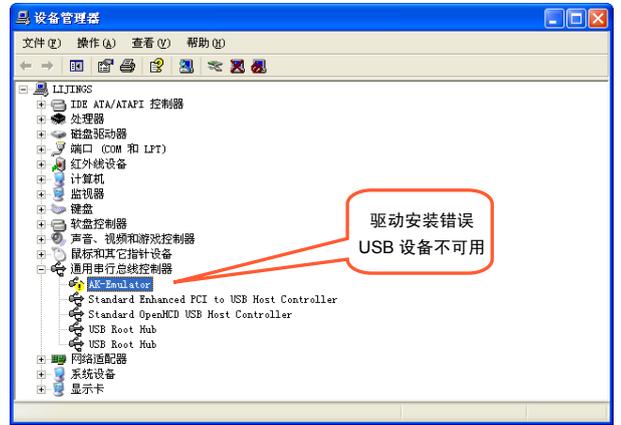


图 2.12 没有正确的安装新硬件结果

2

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。

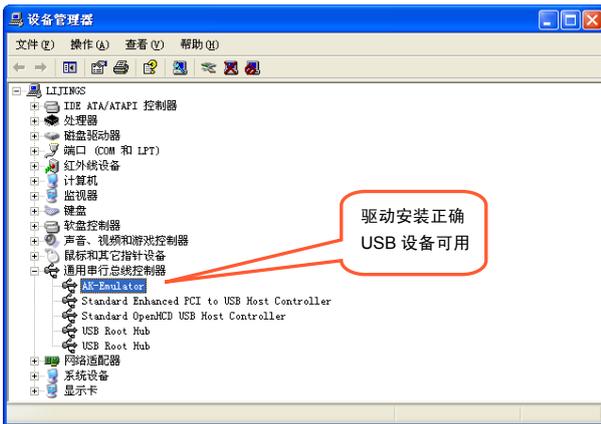


图 2.11 正确的安装新硬件结果

4

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。

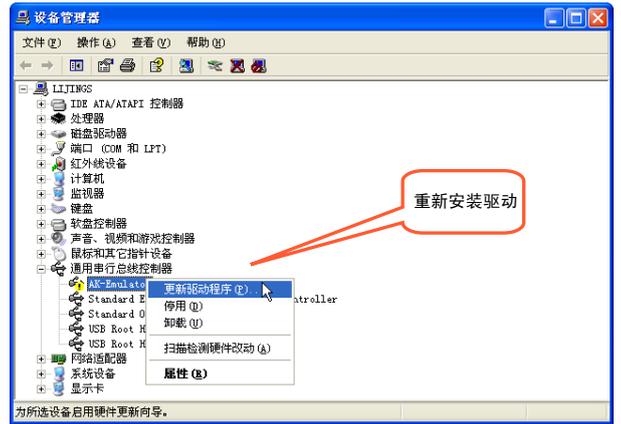
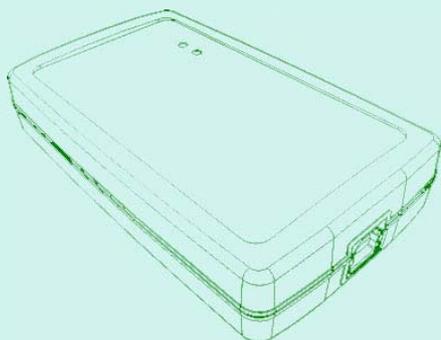


图 2.13 更新驱动程序

## 第 3 章

# Keil RealView MDK 环境 仿真方法



<b>3.1</b>	<b>仿真环境设置</b>	<b>15</b>
<b>3.2</b>	<b>仿真器参数设置</b>	<b>16</b>
3.2.1	硬件选择	16
3.2.2	主要设置	17
3.2.3	TAP 设置	18
3.2.4	程序烧写	19
3.2.5	初始化宏	22
3.2.6	硬件自检	24
<b>3.3</b>	<b>仿真调试</b>	<b>24</b>
3.3.1	开始仿真调试	24
3.3.2	仿真调试工具	25
3.3.3	仿真调试结果	26

## 3.1 仿真环境设置

1 在 Keil RealView MDK 环境下打开一个编译 OK 的工程，如图 3.1 所示。

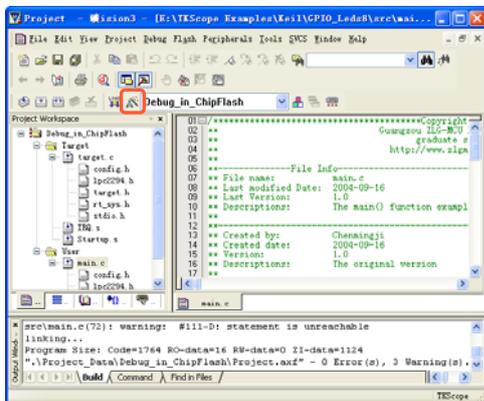


图 3.1 MDK 主界面

2 点击图 3.1 中的  图标，进入如图 3.2 所示的工程设置界面。

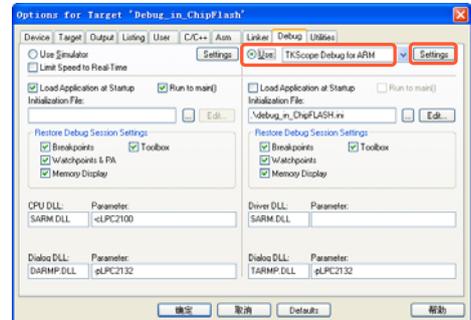


图 3.2 工程设置界面

3 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 AK100 仿真器设置界面，如图 3.3 所示。

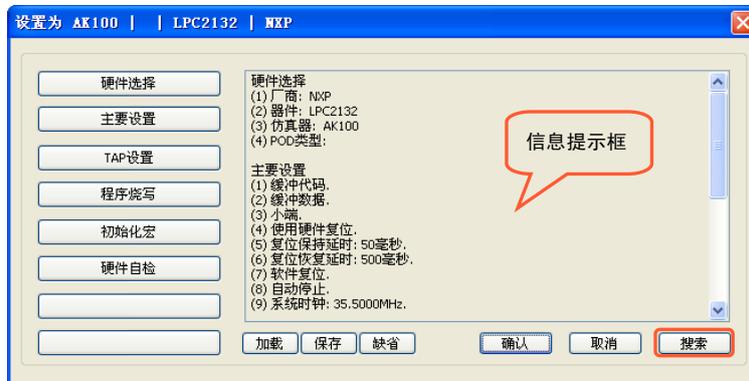


图 3.3 仿真器设置界面



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



AK100 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。

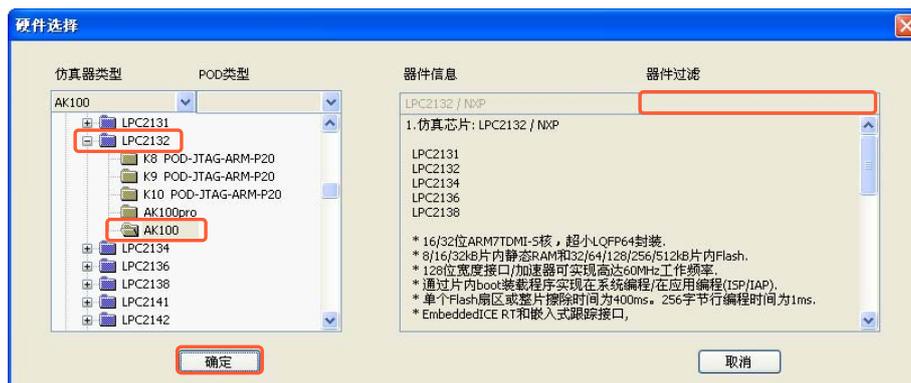


图 3.4 硬件选择界面



用户必须根据实际仿真情况，在【硬件选择】选项的列表中，正确选择芯片型号和仿真器种类。

- 1 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。
- 2 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使用的仿真器种类，如图 3.5 所示。

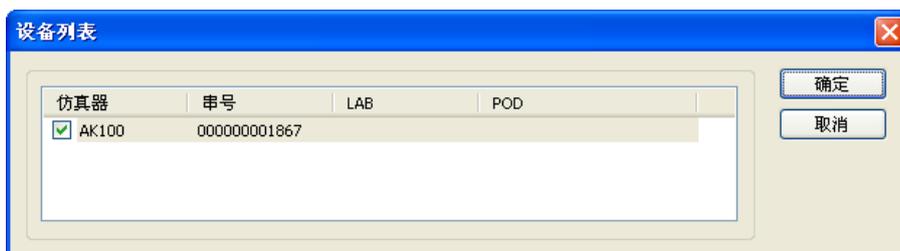


图 3.5 系统搜索结果

## 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。



图 3.6 主要设置界面

### 1 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】。

### 2 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。

### 3 断点

【使用软件断点】：选中此项，则可实现 RAM 中无限断点调试。

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

### 4 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。

### 5 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。

### 6 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，选择后将运行的非常缓慢。

【固定时钟】：选择用户输入的时钟频率数值。

# 7 硬件复位

【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！

# 8 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。
- 【**添加**】：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。

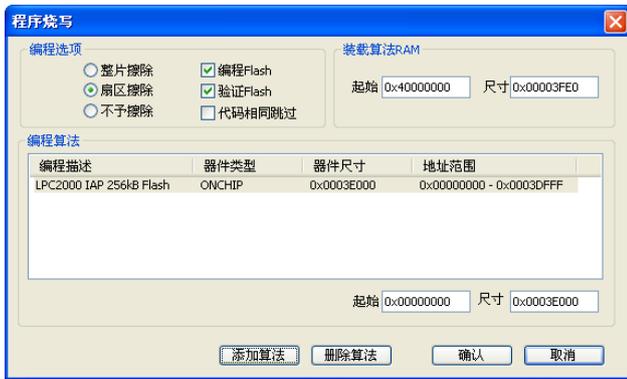


图 3.9 程序烧写界面

## 1 Flash 编程设置

- 【**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。
- 【**不予擦除**】：不进行 Flash 擦除。

## 2 装载算法 RAM

- 【**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。适当减小可加快下载速度。

## 3 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。

## 4 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。

## 1 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。

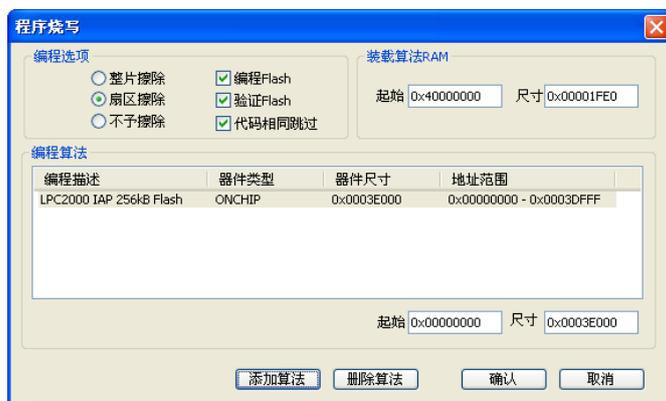


图 3.10 程序烧写界面 1

## 2 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，选择在外部分 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。

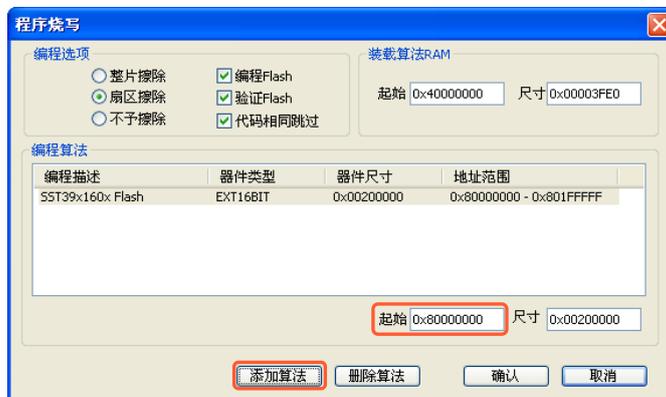


图 3.11 程序烧写界面 2



AK100 仿真器提供众多外部 Flash 的算法文件，存放路径为 AK100 驱动安装目录下 TKScope\configuration\ExtFlash。

### 3 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 0x80000000、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

### 4 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。

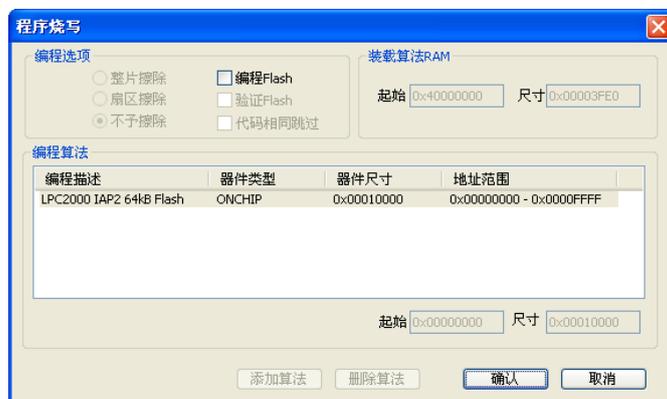


图 3.13 程序烧写界面 4

## 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。

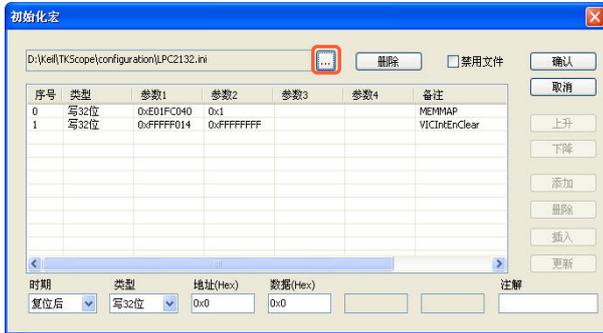


图 3.14 初始化宏界面

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。

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。



AK100 仿真器为数百种芯片提供完善的初始化文件，存放路径为 AK100 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。

表 3.2 现在支持的动作

名称	意义
<b>Read 32bit</b>	读取指定地址的 32 位数据。
<b>Read 16bit</b>	读取指定地址的 16 位数据。
<b>Read 8bit</b>	读取指定地址的 8 位数据。
<b>Write 32bit</b>	在指定地址写入指定的 32 位数据。
<b>Write 16bit</b>	在指定地址写入指定的 16 位数据。
<b>Write 8bit</b>	在指定地址写入指定的 8 位数据。
<b>SetJtagClock</b>	指定 Jtag 工作模式和工作频率（10 进制）。
<b>Delay</b>	延时一段时间，单位为 ms（10 进制）。
<b>SetPC</b>	设置当前程序指针（16 进制）。
<b>Message</b>	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 = "SyncJtagClock"
```

## 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。



图 3.15 硬件自检界面

AK100 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。



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 3.16 所示。

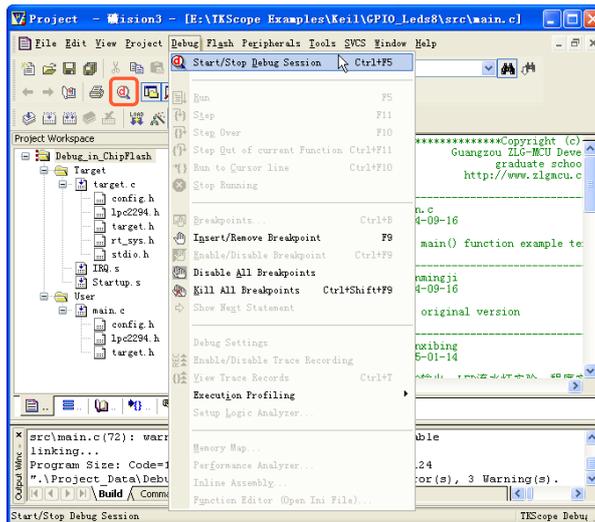


图 3.16 进入调试状态

### 3.3.2 仿真调试工具

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
	进入/退出调试状态 (Start/Stop Debug Session)。
	复位 (Reset CPU)。
	全速运行 (Go)。
	停止运行 (Stop)。
	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。
	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。
	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)。
	取消所有断点 (Kill All Breakpoints)。
	打开/关闭断点 (Enable/Disable Breakpoint)。
	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
	打开反汇编窗口 (Disassembly Window)。
	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。



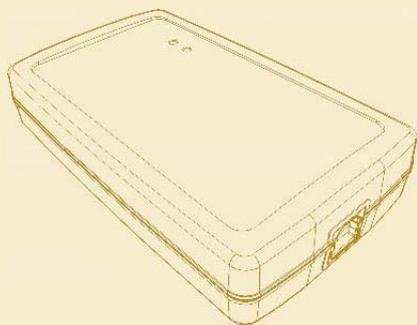
用户如果选择在 Flash 中调试，程序写到芯片的 Flash 中，掉电后会保存，再次上电目标板会执行写入的程序。



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



4.1	添加驱动文件	28
4.2	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

## 4.1 添加驱动文件

1

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。

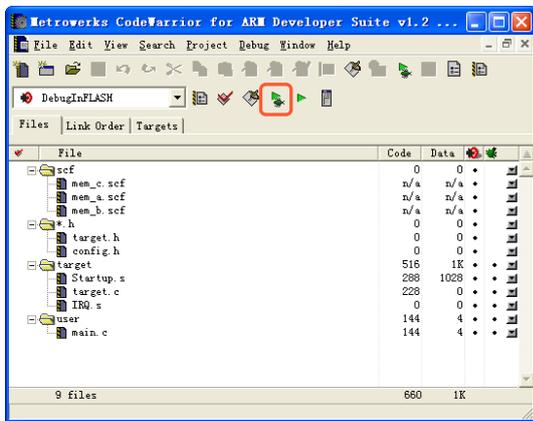


图 4.1 ADS 主界面

2

在 AXD 调试环境下，选择【Options】菜单下【Configure Target】选项，如图 4.2 所示。

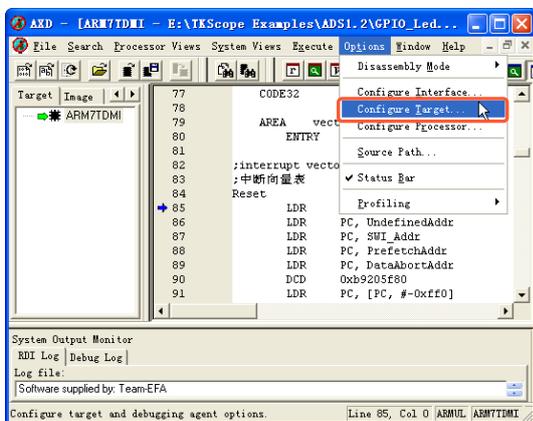


图 4.2 AXD 主界面

3

系统弹出【Choose Target】窗口，如图 4.3 所示。点击图中【Add】按钮，添加 AK100 仿真器的驱动文件。

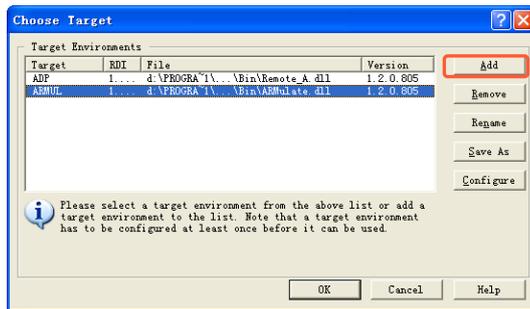


图 4.3 Choose Target 窗口

4

系统弹出驱动选择对话框，打开 AK100 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。

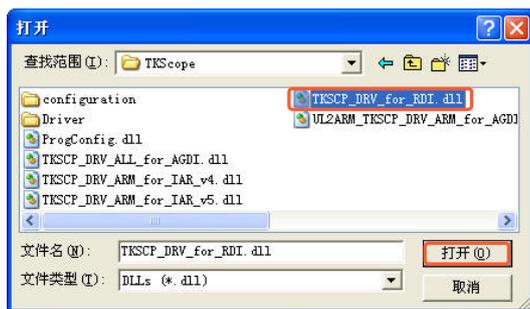


图 4.4 选择驱动文件

5

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 AK100 仿真器的驱动，点击【Configure】按钮，即可进入 AK100 仿真器设置界面，如图 4.6 所示。

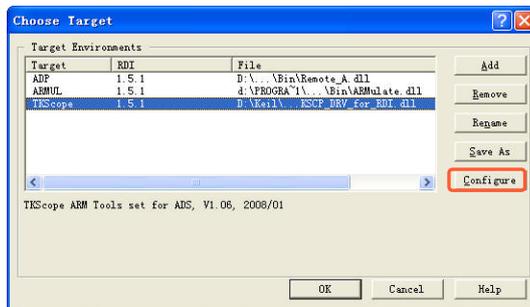


图 4.5 驱动安装完毕

**6** 无论在何种 IDE 环境下，AK100 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



AK100 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 AK100 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)。
	停止运行 (Stop)。
	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。
	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。
	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)。运行程序直到当前光标所在行时停止。
	设置断点 (Toggle BreakPoint)。

表 4.2 AXD 调试观察窗口工具

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

表 4.3 AXD 文件操作工具

快捷图标	意义
	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。



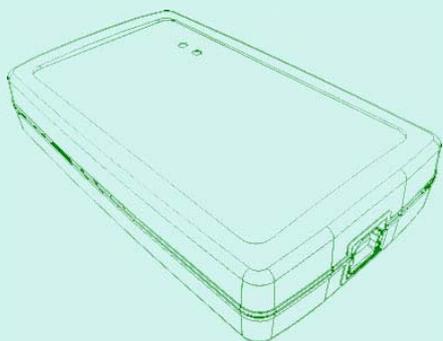
用户如果选择在 Flash 中调试，程序写到芯片的 Flash 中，掉电后会保存，再次上电目标板会执行写入的程序。



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

## 5.1 添加驱动文件

1

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。

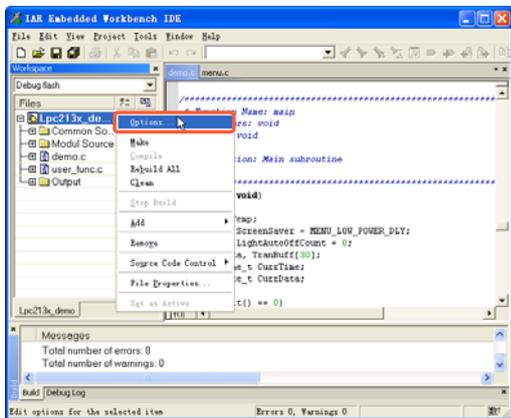


图 5.1 IAR 主界面

2

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】。

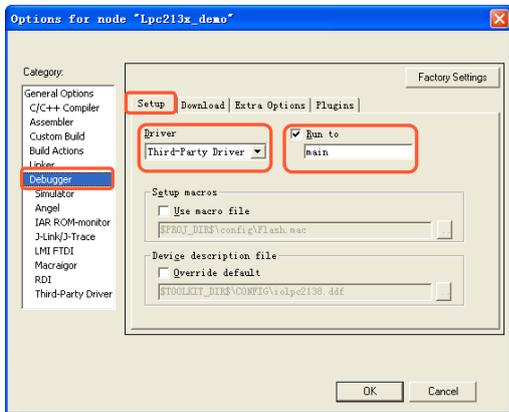


图 5.2 Debugger Setup 界面

3

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。

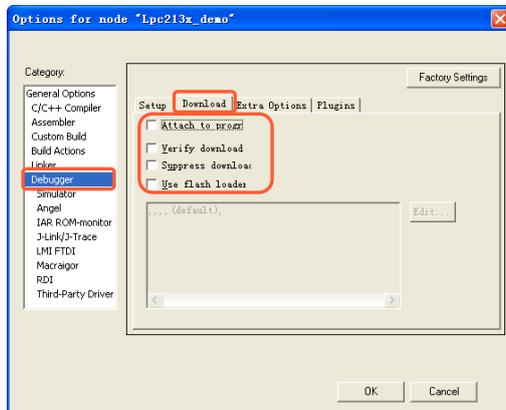


图 5.3 Debugger Download 界面

4

选择【Third-Party Driver】选项，界面如图 5.4 所示。点击图中  图标，添加 AK100 仿真器驱动。

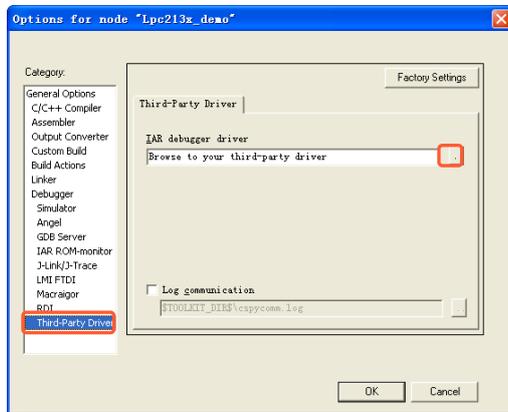


图 5.4 Third-Party Driver 界面

# 5

系统弹出驱动选择对话框，打开 AK100 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。

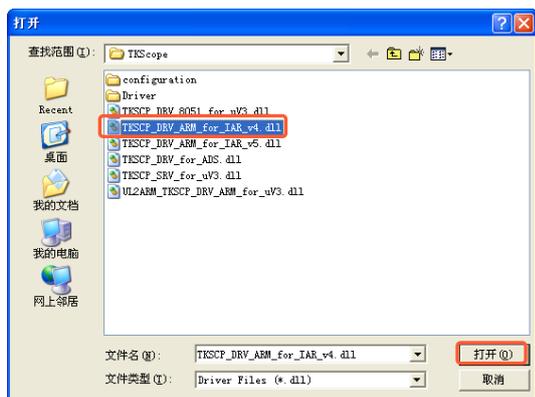


图 5.5 选择驱动文件

# 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。

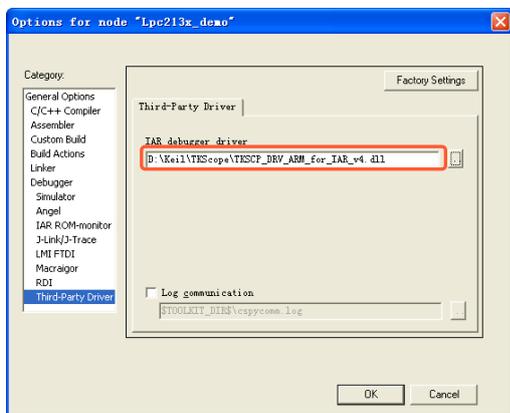


图 5.6 驱动安装完毕

# 7

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。

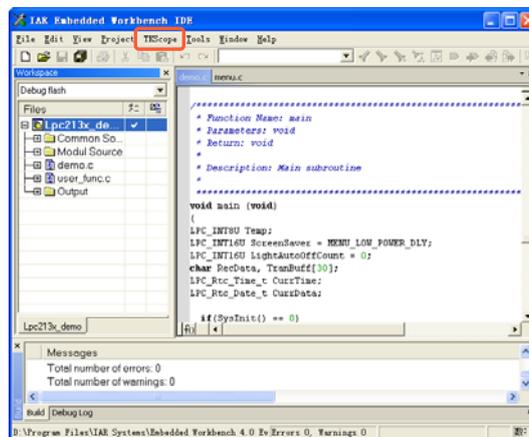


图 5.7 驱动安装完成主界面

# 8

选择【TKScope】菜单下的【Setup】选项，如图 5.8 所示，即可进入 AK100 仿真器设置界面。

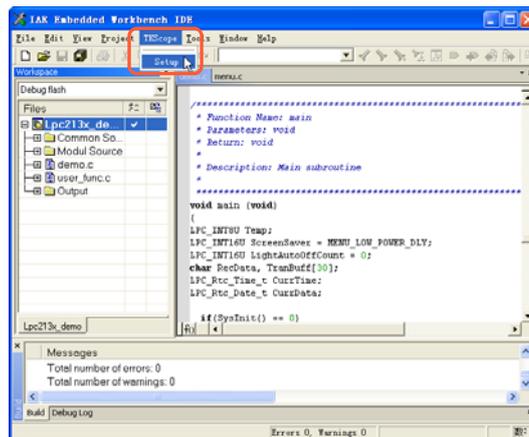


图 5.8 选择 AK100 仿真器设置

9 无论在何种 IDE 环境下，AK100 仿真器设置界面都是一样的（例如，IAR 环境下的图 5.9 与 Keil RealView MDK 环境下的图 3.3），设置方法也是一样的。



图 5.9 仿真器设置界面



AK100 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。

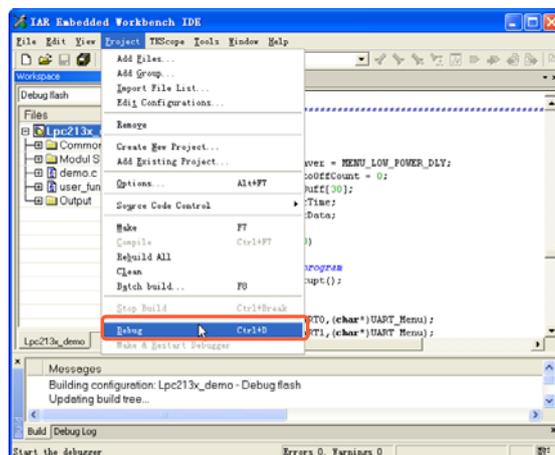


图 5.10 进入调试状态

## 5.2.2 仿真调试工具

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
	编译并调试（Make and Debug）。
	设置断点（Toggle Breakpoint）。
	复位（Reset）。
	停止运行（Break）。
	布越（Step Over）。单步执行程序，跳过子程序，不进入到内部。
	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。
	运行到下一条语句（Next Statement）。程序运行到下一条语句时停止。
	运行到光标（Run to Cursor）。程序运行到当前光标所在行时停止。
	全速运行（Go）。
	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。

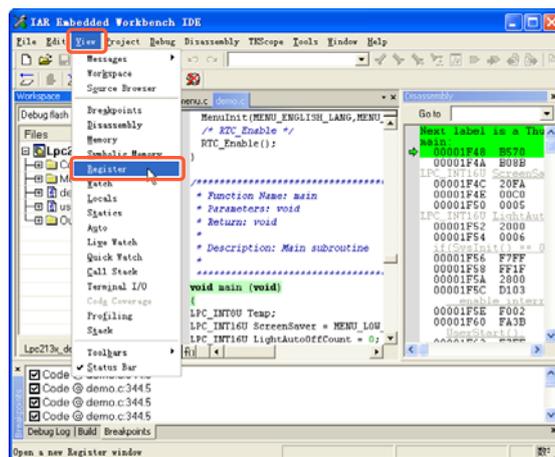


图 5.11 打开观察窗口

### 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。



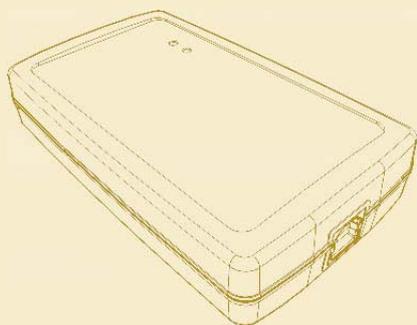
用户如果选择在 Flash 中调试，程序写到芯片的 Flash 中，掉电后会保存，再次上电目标板会执行写入的程序。



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

1

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。

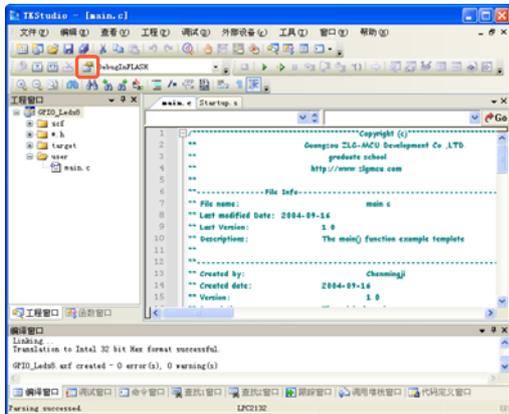


图 6.1 TKStudio 主界面

2

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

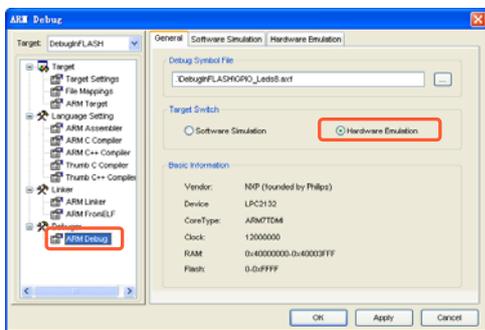


图 6.2 工程配置界面

3

打开【Hardware Emulation】界面，如图 6.3 所示。点击【Add】按钮，添加 AK100 仿真器驱动文件。

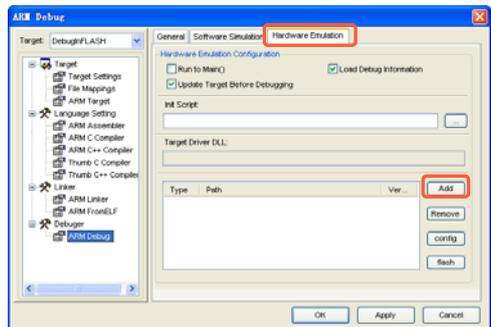


图 6.3 硬件仿真设置窗口

4

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】。

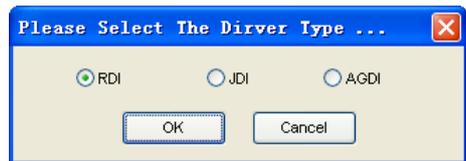


图 6.4 驱动类型选择

5

打开 AK100 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。

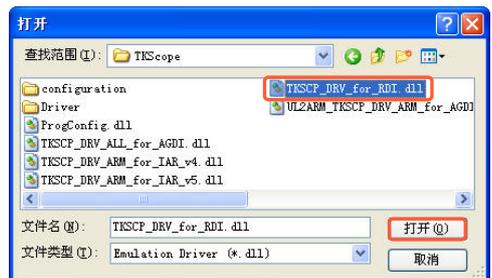


图 6.5 选择驱动文件

6 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 AK100 仿真器的驱动，点击【Config】按钮，即可进入 AK100 仿真器设置界面，如图 6.7 所示。

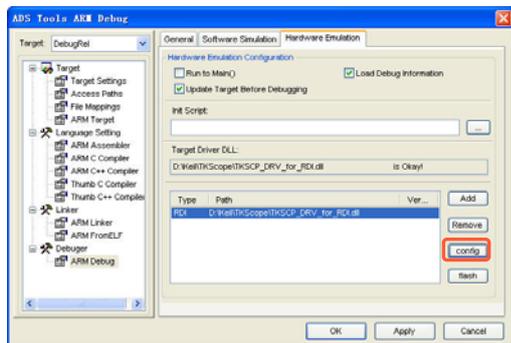


图 6.6 驱动安装完毕

7 无论在何种 IDE 环境下，AK100 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。



图 6.7 仿真器设置界面



AK100 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。

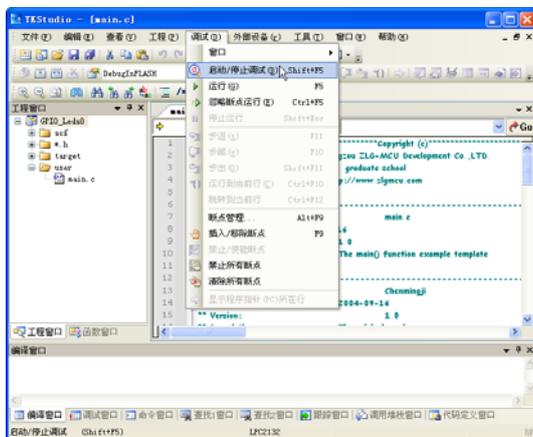


图 6.8 进入调试状态

## 6.2.2 仿真调试工具

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
	启动/停止调试。
	复位。
	全速运行。
	忽略断点运行。
	停止运行。
	单步。单步执行程序，进入到子程序内部。
	布越。单步执行程序，跳过子程序，不进入到内部。
	步出。执行到当前子函数的结束。
	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。
	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
	显示/隐藏反汇编窗口。
	观察变量窗口。
	观察存储器窗口。
	观察寄存器窗口。
	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。



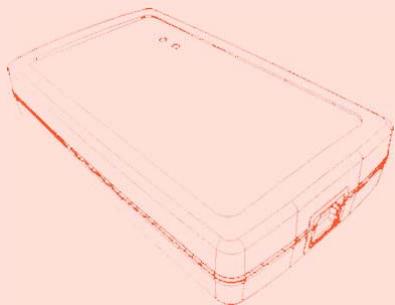
用户如果选择在 Flash 中调试，程序写到芯片的 Flash 中，掉电后会保存，再次上电目标板会执行写入的程序。



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

# AK100 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，AK100 仿真器会不断升级，支持新型号器件的仿真。AK100 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。

AK100 仿真器的驱动程序公布在官方网站上 <http://www.zlqmcu.com>，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 AK100 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。

### 1 电话（传真）

使用电话沟通交流是最快的解决问题的途径。请您在拨打电话之前，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问题。

技术支持 Email: [TKS@zlqmcu.com](mailto:TKS@zlqmcu.com)。

### 3 BBS

欢迎访问周立功单片机论坛

<http://www.zlqmcu.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。

### 4 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

AK100 仿真器是广州致远电子有限公司在 2009 年隆重推出的一款高性能 ARM 专用仿真器，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。

AK100 仿真器可以在 Keil/ADS/IAR/TKStudio 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，这些优质的性能无疑让其成为 ARM 仿真器中的明星产品，必将引领 ARM 仿真的新潮流。

再次感谢您选用 AK100 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 <http://www.zlgmcu.com>。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 AK100 仿真器支持芯片列表

AK100 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，AK100 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。

## ■ Actel

### Cortex-M3 内核

A2F200M3A, A2F200M3B, A2F200M3C, A2F200M3D, A2F200M3E, A2F200M3F, A2F500M3A, A2F500M3B, A2F500M3C, A2F500M3D, A2F500M3E, A2F500M3F, A2F500M3G  
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

**ARM 内核** ADuC7019BCPZ62, ADuC7020BCPZ62, ADuC7021BCPZ62, ADuC7021BCPZ32, ADuC7022BCPZ32, ADuC7022BCPZ62, ADuC7023BCPZ62I, ADuC7024BCPZ62, ADuC7025BCPZ62, ADuC7025BCPZ32, ADuC7026BSTZ62, ADuC7027BSTZ62, ADuC7028BBCZ62, ADuC7029BBCZ62, ADUC7030, ADUC7032, ADUC7033, ADUC7034, ADUC7036, ADUC7037, ADUC7038, ADUC7039, ADUC7060, ADUC7061, ADUC7062, ADUC7121, ADUC7122, ADUC7124, ADUC7126, ADUC7128, ADUC7129  
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

**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

**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

## ■ EnergyMicro

### Cortex-M3 内核

#### Gecko 系列

EFM32G200F16, EFM32G200F32, EFM32G200F64, EFM32G210F128, EFM32G222F128, EFM32G222F32, EFM32G222F64, EFM32G230F128, EFM32G230F32, EFM32G230F64, EFM32G232F128, EFM32G232F32, EFM32G232F64, EFM32G280F128, EFM32G280F32, EFM32G280F64, EFM32G290F128, EFM32G290F32, EFM32G290F64, EFM32G840F128, EFM32G840F32, EFM32G840F64, EFM32G842F128, EFM32G842F32, EFM32G842F64, EFM32G880F128, EFM32G880F32, EFM32G880F64, EFM32G890F128, EFM32G890F32, EFM32G890F64

#### Giant Gecko 系列

EFM32GG230F1024, EFM32GG230F512, EFM32GG232F1024, EFM32GG232F512, EFM32GG280F1024, EFM32GG280F512, EFM32GG290F1024, EFM32GG290F512, EFM32GG295F1024, EFM32GG295F512, EFM32GG330F1024, EFM32GG330F512, EFM32GG332F1024, EFM32GG332F512, EFM32GG380F1024, EFM32GG380F512, EFM32GG390F1024, EFM32GG390F512, EFM32GG395F1024, EFM32GG395F512, EFM32GG840F1024, EFM32GG840F512, EFM32GG842F1024, EFM32GG842F512, EFM32GG880F1024, EFM32GG880F512, EFM32GG890F1024, EFM32GG890F512, EFM32GG895F1024, EFM32GG895F512, EFM32GG940F1024, EFM32GG940F512, EFM32GG942F1024, EFM32GG942F512, EFM32GG980F1024, EFM32GG980F512, EFM32GG990F1024, EFM32GG990F512, EFM32GG995F1024, EFM32GG995F512

#### Leopard Gecko 系列

EFM32LG230F128, EFM32LG230F256, EFM32LG230F64, EFM32LG232F128, EFM32LG232F256, EFM32LG232F64, EFM32LG280F128, EFM32LG280F256, EFM32LG280F64, EFM32LG290F128, EFM32LG290F256, EFM32LG290F64, EFM32LG295F128, EFM32LG295F256, EFM32LG295F64, EFM32LG330F128, EFM32LG330F256, EFM32LG330F64, EFM32LG332F128, EFM32LG332F256, EFM32LG332F64, EFM32LG380F128, EFM32LG380F256, EFM32LG380F64, EFM32LG390F128, EFM32LG390F256, EFM32LG390F64, EFM32LG395F128, EFM32LG395F256, EFM32LG395F64, EFM32LG840F128, EFM32LG840F256, EFM32LG840F64, EFM32LG842F128, EFM32LG842F256, EFM32LG842F64, EFM32LG880F128, EFM32LG880F256, EFM32LG880F64, EFM32LG890F128, EFM32LG890F256, EFM32LG890F64, EFM32LG895F128, EFM32LG895F256, EFM32LG895F64, EFM32LG940F128, EFM32LG940F256, EFM32LG940F64, EFM32LG942F128, EFM32LG942F256, EFM32LG942F64, EFM32LG980F128, EFM32LG980F256, EFM32LG980F64, EFM32LG990F128, EFM32LG990F256, EFM32LG990F64, EFM32LG995F128, EFM32LG995F256, EFM32LG995F64

#### Tiny Gecko 系列

EFM32TG108F16, EFM32TG108F32, EFM32TG108F4, EFM32TG108F8, EFM32TG110F16, EFM32TG110F32, EFM32TG110F4, EFM32TG110F8, EFM32TG210F16, EFM32TG210F32, EFM32TG210F8, EFM32TG222F16, EFM32TG222F32, EFM32TG222F8, EFM32TG230F16, EFM32TG230F32, EFM32TG230F8, EFM32TG232F16, EFM32TG232F32, EFM32TG232F8, EFM32TG822F16, EFM32TG822F32, EFM32TG822F8, EFM32TG840F16, EFM32TG840F32, EFM32TG840F8, EFM32TG842F16, EFM32TG842F32, EFM32TG842F8

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, MAC7142

其它型号芯片陆续支持

## ■ Fujitsu

### Cortex-M3 内核

MB9AF102NA, MB9AF104NA, MB9AF102RA, MB9AF104RA, MB9AF111L, MB9AF112L, MB9AF114L, MB9AF111M, MB9AF112M, MB9AF114M, MB9AF115M, MB9AF116M, MB9AF111N, MB9AF112N, MB9AF114N, MB9AF115N,

MB9AF116N, MB9AF131K, MB9AF132K, MB9AF131L, MB9AF132L, MB9AF311L, MB9AF312L, MB9AF314L, MB9AF311M, MB9AF312M, MB9AF314M, MB9AF315M, MB9AF316M, MB9AF311N, MB9AF312N, MB9AF314N, MB9AF315N, MB9AF316N, MB9BF104NA, MB9BF105NA, MB9BF106NA, MB9BF104RA, MB9BF105RA, MB9BF106RA, MB9BF116S, MB9BF117S, MB9BF118S, MB9BF116T, MB9BF117T, MB9BF118T, MB9BF216S, MB9BF217S, MB9BF218S, MB9BF216T, MB9BF217T, MB9BF218T, MB9BF304NA, MB9BF305NA, MB9BF306NA, MB9BF304RA, MB9BF305RA, MB9BF306RA, MB9BF316S, MB9BF317S, MB9BF318S, MB9BF316T, MB9BF317T, MB9BF318T, MB9BF404NA, MB9BF405NA, MB9BF406NA, MB9BF404RA, MB9BF405RA, MB9BF406RA, MB9BF416S, MB9BF417S, MB9BF418S, MB9BF416T, MB9BF417T, MB9BF418T, MB9BF504NA, MB9BF505NA, MB9BF506NA, MB9BF504RA, MB9BF505RA, MB9BF506RA, MB9BF516S, MB9BF517S, MB9BF518S, MB9BF516T, MB9BF517T, MB9BF518T, MB9BF616S, MB9BF617S, MB9BF618S, MB9BF616T, MB9BF617T, MB9BF618T  
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

## ■ MARVELL (Intel)

**XSCALE 内核**

PXA255, PXA270

其它型号芯片陆续支持

## ■ NUVOTON (WinBOND)

**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, NUC120LD2AN, NUC120LD1AN, NUC120LC1AN, NUC120RD2AN, NUC120RD1AN, NUC120RC1AN, NUC130LE3AN, NUC130LD3AN, NUC130LD2AN, NUC130RE3AN, NUC130RD3AN, NUC130RD2AN, NUC130VE3AN, NUC130VD3AN, NUC130VD2AN, NUC140LE3AN, NUC140LD3AN, NUC140LD2AN, NUC140RE3AN, NUC140RD3AN, NUC140RD2AN, NUC140VE3AN, NUC140VD3AN, NUC140VD2AN

其它型号芯片陆续支持

**Flash 器件** W19B320AB, W19B320AT, W19B320BB, W19B320BT

其它型号 Flash 器件陆续支持

## ■ NXP

**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, LPC11U12x201, LPC11U13x201, LPC11U14x201, LPC1102, LPC1111x102, LPC1111x202, LPC1112x102, LPC1112x202, LPC1113x202, LPC1113x302, LPC1114x202, LPC1114x302, LPC11C12x301, LPC11C14x301, LPC11C22x301, LPC11C24x301, LPC11U12x201, LPC11U12x201, LPC11U13x201, LPC11U14x201, LPC11U14x201, LPC11U14x201, LPC11U23x301, LPC11U24x/301, LPC11U24x401, LPC11U24x401, LPC11U24x301, LPC11U24x301, LPC11U24x401, LPC1224x101, LPC1224x121, LPC1225x301, LPC1225x321, LPC1226x301, LPC1227x301

其它型号芯片陆续支持

## ■ OKI

**ARM 内核** 全系列 ARM 内核，即将支持

## ■ Samsung

**ARM 内核** S3C44B0X, S3C2410A, S3C2416X, S3C2440A, S3C4510

其它型号芯片陆续支持

## ■ SST

**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

STM32F100C4, STM32F100R4, STM32F100C6, STM32F100R6, STM32F100C8, STM32F100R8, STM32F100V8, STM32F100CB, STM32F100RB, STM32F100VB, STM32F100RC, STM32F100VC, STM32F100ZC, STM32F100RD, STM32F100VD, STM32F100ZD, STM32F100RE, STM32F100VE, STM32F100ZE, STM32F101C4, STM32F101R4, STM32F101T4, STM32F101C6, STM32F101R6, STM32F101T6, STM32F101C8, STM32F101R8, STM32F101V8, STM32F101T8, STM32F101RB, STM32F101VB, STM32F101CB, STM32F101RC, STM32F101VC, STM32F101ZC, STM32F101RD, STM32F101VD, STM32F101ZD, STM32F101RE, STM32F101VE, STM32F101ZE, STM32F101RF, STM32F101VF, STM32F101VG, STM32F101ZG, STM32F102C4, STM32F102R4, STM32F102C6, STM32F102R6, STM32F102C8, STM32F102R8, STM32F102CB, STM32F102RB, STM32F103C4, STM32F103R4, STM32F103T4, STM32F103C6, STM32F103R6, STM32F103T6, STM32F103C8, STM32F103R8, STM32F103V8, STM32F103T8, STM32F103RB, STM32F103VB, STM32F103CB, STM32F103TB, STM32F103RC, STM32F103VC, STM32F103ZC,

STM32F103RD, STM32F103VD, STM32F103ZD, STM32F103RE, STM32F103VE, STM32F103ZE, STM32F103RF, STM32F103VF, STM32F103ZF, STM32F103RG, STM32F103VG, STM32F103ZG, STM32F105R8, STM32F105V8, STM32F105RB, STM32F105VB, STM32F105RC, STM32F105VC, STM32F107RB, STM32F107VB, STM32F107RC, STM32F107VC, STM32F205RB, STM32F205VB, STM32F205RC, STM32F205VC, STM32F205ZC, STM32F205RE, STM32F205VE, STM32F205ZE, STM32F205RF, STM32F205VF, STM32F205ZF, STM32F205RG, STM32F205VG, STM32F205ZG, STM32F207IC, STM32F207VC, STM32F207ZC, STM32F207IE, STM32F207VE, STM32F207ZE, STM32F207IF, STM32F207VF, STM32F207ZF, STM32F207IG, STM32F207VG, STM32F207ZG, STM32F215RE, STM32F215VE, STM32F215ZE, STM32F215RG, STM32F215VG, STM32F215ZG, STM32F217IE, STM32F217VE, STM32F217ZE, STM32F217IG, STM32F217VG, STM32F217ZG, STM32L151C8, STM32L151R8, STM32L151V8, STM32L151CB, STM32L151RB, STM32L151VB, STM32L152C8, STM32L152R8, STM32L152V8, STM32L152CB, STM32L152RB, STM32L152VB, STM32W108C8, STM32W108CB, STM32W108CC, STM32W108CZ, STM32W108HB  
其它型号芯片陆续支持

## ■ TI

**ARM 内核** TMS470R1A64, TMS470R1A128, TMS470R1A256  
其它型号芯片陆续支持

### 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, LM3S1332, LM3S1435, LM3S1439, LM3S1512, LM3S1538, LM3S1601, LM3S1607, LM3S1608, LM3S1620, LM3S1625, LM3S1626, LM3S1627, LM3S1635, LM3S1637, LM3S1751, LM3S1776, LM3S1850, 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, LM3S2911, LM3S2918, LM3S2939, LM3S2948, LM3S2950, LM3S2965, LM3S3739, LM3S3748, LM3S3749, LM3S3651, LM3S3759, LM3S3768, LM3S5632, LM3S5732, LM3S5737, LM3S5739, LM3S5747, LM3S5749, LM3S5652, LM3S5662, LM3S5752, LM3S5757, LM3S5762, LM3S5767, LM3S5768, LM3S5769, LM3S6100, LM3S6110, LM3S6420, LM3S6422, LM3S6432, LM3S6537, LM3S6610, LM3S6611, LM3S6618, LM3S6633, LM3S6637, LM3S6730, LM3S6753, LM3S6816, LM3S6911, LM3S6916, 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

## ■ TOSHIBA

### Cortex-M3 内核

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