

# EasyARM-LPC54018 快速入门指

EasyARM-LPC54018 快速入门指南

UM01010101 V1.01 Date:

Date: 2019/11/11

产品入门手册

| 类别  | 内容           |
|-----|--------------|
| 关键词 | 开发资料介绍,开发板介绍 |
| 摘要  | 开发板介绍和资料结构   |





EasyARM-LPC54018 入门

修订历史

| 版本    | 日期         | 原因   |
|-------|------------|------|
| V1.00 | 2019/10/15 | 创建文档 |
| V1.01 | 2019/11/11 | 修改文档 |



目 录

| 1. | EasyARM- | LPC54018 开发套件简介             | 1  |
|----|----------|-----------------------------|----|
|    | 1.1 L    | PC54018 微控制器简介              | 1  |
|    | 1.1.1    | LPC54018 性能特色               | 1  |
|    | 1.2 E    | asyARM-LPC54018             | 2  |
| 2. | 资料日录约    |                             | 3  |
| 3  | 例程使用非    |                             | 5  |
| 5. | 3.1 L    | ED 例程                       | 5  |
|    | 311      | 程序结构说明                      | 5  |
|    | 312      | 在线调试                        | 6  |
|    | 3.1.2    | μス码风<br>使田 I_I ink 下裁离线调试程序 | 8  |
| 4. | 免责声明     | [X/1] J-LIIIK   刊刊中以11/1    | 14 |

# 1. EasyARM-LPC54018 开发套件简介

### 1.1 LPC54018 微控制器简介

LPC540xx 属于面向嵌入式应用的 ARM Cortex-M4 微控制器系列,具有丰富的外设集, 极低的功耗和增强的调试功能。

ARM Cortex-M4 是一款 32 位内核,可提供诸如低功耗,增强的调试功能,以及高水平的支持块集成。ARM Cortex-M4 CPU 集成了三级流水线,采用哈佛架构,具有单独的本地指令和数据总线以及外围设备的第三个总线,并包含支持推测分支的内部预取单元。ARM Cortex-M4 支持单周期数字信号处理和 SIMD 指令。硬件浮点处理器集成到内核中。

LPC540xx 系列包含 360KB 的片上 SRAM, 一个用以扩展程序存储器的 quad SPI 闪存接口(SPIFI),一个高速和一个全速 USB 主机和设备控制器,以太网 AVB, LCD 控制器,智能卡接口,SD/MMC, CAN FD,一个外部存储器控制器(EMC),一个带 PDM 麦克风接口和 I<sup>2</sup>S 的 DMIC 子系统,五个通用定时器,SCTimer/PWM,RTC/报警定时器,多速率定时器(MRT),窗口看门狗定时器(WWDT),十个灵活的串行通信外设(USART,SPI,I<sup>2</sup>S,I<sup>2</sup>C 接口),12 位 5.0 Msamples / sec ADC,温度传感器,1.71V 至 3.6V 单电源,AES-256和安全散列算法(SHA)。

#### 1.1.1 LPC54018 性能特色

- ▶ 采用 32 位的具备浮点单元和存储器保护单元的 ARM Cortex-M4 内核,运行频率高达 180 MHz;
- ▶ 高达 360KB 的 SRAM (包含 160 KB 连续的主 SRAM 和一个 I&D 总线上的额外 32 KB SRAM, 8 KB SRAM 块用于 USB 流量);
- ▶ 支持 ROM API, 闪存在应用编程(IAP)和在系统编程(ISP),可通过闪存、USART、 SPI和I℃中的有效用户代码进行启动;
- ➢ Flexcomm 接口包含 10 个串行外设。每个 Flexcomm 接口可通过软件选择作为 USART、SPI 或 I<sup>2</sup>C 接口;
- ▶ I<sup>2</sup>C 总线接口支持 Fast-Mode 和 Fast-Mode Plus,数据传输率高达 1Mbit/S,具有多种 地址识别和监控模式,两组真正的 I2C 焊盘,支持从设备高速模式 (3.4Mbit/S);
- ▶ LCD 控制器中有一个专用的 DMA 控制器,可选的显示分辨率最高达到 1024 x 768 像素,支持 24 位真彩模式;
- ▶ 外部存储控制器(EMC)支持异步静态存储器设备,如RAM、ROM、闪存以及动态存储器,如单数据速率SDRAM,其SDRAM时钟最高达100MHz;
- ▶ 安全散列算法(SHA1/SHA2)模块支持通过专用 DMA 控制器进行安全引导,增强的代码读取保护(eCRP)用于保护用户代码;
- 可编程 PMU(电源管理单元)用于尽可能降低功耗,从而匹配不同性能水平的需求; 具备睡眠、深度睡眠和深度节电模式3中低功耗模式;
- ▶ 宽电源范围 1.71V 至 3.6V;
- ▶ 外界的正常工作温度在-40℃到 105℃之间;
- ▶ 处理器提供 TFBGA180、TFBGA100、LQFP 和 LQFP208 封装,拥有最多高达 171 个通用输入/输出(GPIO)引脚;



# 1.2 EasyARM-LPC54018

EasyARM-LPC54018开发套件主要用于 LPC540xx 系列微控制器的学习和开发。

EasyARM-LPC54018 开发板基于 NXP 半导体的 LPC54018JET180 微控制器。在一块核 心板上电路板上包含了 1 路标准的 AWorks 接口、1 路 MicroPort 接口、1 路 MiniPort 接口、1 路 LCD 接口、1 路以太网接口、2 路 USB 接口(每一路包含 USB Host 和 USB Device 接口) 和 1 路 EMC 扩展接口,如图 1.1。这些接口不仅把单片机的所有 I/O 资源引出,还可以借助 这些接口外扩多种模块。板上资源包含 SDRAM, SPI Flash, 2 个 LED 发光二极管、1 个无 源蜂鸣器、1 个加热电阻、1 个 LM75B 测温芯片、1 个热敏电阻、1 个 TL431 基准源、1 个 多功能按键和 1 个复位按键,可以完成多种基础实验。例如:加热电阻配合 LM75B 和热敏 电阻可实现数字和模拟测温。



#### 图 1.1 EasyARM-LPC54018 开发板正面图片

EasyARM-LPC54018 开发套件的出现简化了用户的硬件设计,使得学习 LPC54018 的难 度和时间大大降低,让初学者也可以很快掌握 32 位 ARM Cortex-M4 内核的开发。

# 2. 资料目录结构说明

开发板提供以下资料:

| 资料          | 所在目录             |
|-------------|------------------|
| 开发板图片       | 1.开发板图片          |
| 快速入门指导      | 2.快速入门           |
| 相关集成开发环境的插件 | 3.集成开发环境         |
| 例程代码        | 4.示例代码           |
| 原理图和硬件设计指南  | 5.硬件设计           |
| 芯片手册        | 6.数据 用户(参考) 勘误手册 |
| 其他          | 7.其他             |

"2.快速入门"中包含使用 EasyARM-LPC54018 开发板的入门指导文档。

"3.集成开发环境"中包含了开发 EasyARM-LPC54018 所需要的 Keil MDK 的 pack 包,用 户亦可在 http://mcuxpresso.nxp.com/cmsis\_pack/repo/NXP.LPC54018M\_DFP.12.0.0.pack 下载。

"4. 例程代码"包含了原厂提供的 SDK 包及周立功公司针对 EasyARM-LPC54018 修改的 SDK 包。如图 2.1 所示是 SDK 包的目录结构。



#### 图 2.1 SDK 包目录结构

SDK 包中"board"目录中是针对不同开发板的例程。如图 2.2 所示是针对 EasyARM-LPC54018开发板的例程目录。其中"driver\_examples"目录中的例程是针对芯片每 个模块的单独例程,"emwin examples"目录中是 emWin 的例程,该例程需要 LCD 屏 幕, "rtos\_examples"目录中是 FreeRTOS 的例程, "usb\_examples"目录中是 USB 的例程, 包 括 USB 设备和 USB 主机的例程;

州立功科技股份有限公司 Easy

EasyARM-LPC54018 快速入门指南 EasyARM-LPC54018 入门



图 2.2 boards

"CMSIS"目录中是 CMSIS 规范相关的头文件和库文件;

"devices"目录中是 LPC546xx 系列芯片的外设驱动库以及不同开发环境(Keil MDK、MCUXpressore、IAR 等)的启动代码;

"docs"目录是 NXP 原厂提供 SDK 包使用的指导文档,如果需要详细了解 SDK 包的使用,可以参考该目录下的文档;

"middleware"目录中包含了常用的中间件,有 USB 协议栈、lwip 协议栈、SD/MMC 协议、FATFS 文件系统的源码,以及 emWin 的库文件和头文件;

"rtos"目录中是实时操作系统,包含 FreeRTOS 的源码;

"tools"目录中是 cmake 工具需要的相关文件。

产品入门手册



# 3. 例程使用指南

这里以 SDK 包中目录" boards\lpcxpresso54018\driver\_examples\gpio"中的例程为例介绍 如何使用 SDK 包中的例程。

## 3.1 LED 例程

boards\lpcxpresso54018\driver\_examples\gpio 目录下的 led\_output 是 LED 的例程。 该例程使用到了 LED1,在线调试该例程前请短接 J10, J14(KEY->PIO1\_22)。

#### 3.1.1 程序结构说明

使用 Keil MDK 打开工程文件" boards\lpcxpresso54018\driver\_examples\gpio\led\_output \mdk \gpio\_led\_output.uvprojx", 工程的结构如图 3.1 所示。

| T 🕅 WarkSusse                |
|------------------------------|
|                              |
| 🗄 🍂 Project: gpio_led_output |
| 🖨 ᇶ gpio_led_output debug    |
| 🖃 🦾 source                   |
| 🖶 📄 gpio_led_output.c        |
| 🖃 🦾 board                    |
| 🕀 📄 pin_mux.c                |
| pin_mux.h                    |
| ⊕ · Doard.c                  |
| board.h                      |
| 🗄 📄 clock_config.c           |
| clock_config.h               |
| doc                          |
| readme.txt                   |
|                              |
| device                       |
|                              |
| component-uart               |
| component-serial_manager     |
| component-lists              |
| 🗈 🔁 startup                  |
| 🕀 🔁 CMSIS                    |
|                              |

图 3.1 工程结构

其中 startup 目录中是芯片的启动代码和系统初始化代码; doc 目录下是例程的说明文件; drivers 目录是芯片的驱动代码,包括芯片所有片上模块的驱动; source 目录是应用程序代码,main 函数位于该目录的 gpio\_led\_output.c 中; board 目录是针对特定板子的引脚复用配置代码(pin\_mux.c pin\_mux.h)、时钟配置代码

```
产品入门手册
```

©2019 ZLG Technology Corp.,Ltd.



(clock\_config.c clock\_config.h)以及板级初始化代码(board.c board.h)。

#### 3.1.2 在线调试

打开工程后,选择在线调试方式,如图 3.2 红框所示,选择 gpio\_led\_output debug。

| 🔣 F:\SVN_other\LPC54018快     | 速入门\SDK\boards\lpcxpresso54018\driver_examples\gp |
|------------------------------|---|
| File Edit View Project Fl    | lash Debug Peripherals Tools SVCS Window Help     |
| D 😂 🖬 🖉   🖇 🖻 🕻              | 3   ウ ♡   ← →   作 務 務 務   澤 澤 /// ///             |
| 🗇 🕮 🛍 🧼 - 🔜   🛱              | gpio_led_output debug 🗔 🕺 着 🗟 🗇 🍏                 |
| Project                      | apio led output debug                             |
| 🖃 🔧 Proiect: apio led output |   |

图 3.2 调试方式选择

选择好调试方式后,进行对工程的设置。点击如图 3.3 红框所示 ▲图标进入工程设置页面。

| )18     | 快速入   | .ï]∖sdk'  | boards\lp                | cxpr | esso5 | 4018\dr    | iver_exa | an |
|---------|-------|-----------|--------------------------|------|-------|------------|----------|----|
| ct      | Flash | Debug     | Peripher                 | als  | Tools | SVCS       | Windo    | w  |
| b       | 8     | 9 6       | $\leftarrow \Rightarrow$ | P    | 12    | P. P.      | *        | Ē  |
| LC<br>J | 🕴 gi  | pio_led_( | output debu              | ıg 🔻 | *     | <b>å</b> 5 | 🔶 🕎      | 2  |

图 3.3 工程设置

在线调试工程时,工程设置只需要设置 Debug 选项卡。如图 3.4 红框所示,调试工具选择为 J-LINK / J-TRACE Cortex。然后点击 Setting 进行相应的 J-LINK 设置。

| 🕎 Options for Target 'gpio_led_output debug'  | ×  |
|---|--|
| Device   Target   Output   Listing   User   C/C++   J   | Asm Linker Debug Utilities   |
| C Use Simulator <u>with restrictions</u> Settings   | Use: J-LINK / J-TRACE Cortex      Settings                         |
| ✓ Load Application at Startup     ✓ Run to main() Initialization File:  | Load Application at Startup     Run to main() Initialization File: |
| Restore Debug Session Settings<br>Breakpoints I Toolbox<br>Watch Windows & Performance Analyzer<br>Memory Display I System Viewer | Restore Debug Session Settings                                     |
| CPU DLL: Parameter:<br>SARMCM3.DLL -REMAP -MPU  | Driver DLL: Parameter:<br>SARMCM3.DLL -MPU                         |
| Dialog DLL: Parameter:<br>DCM.DLL -pCM4   | Dialog DLL: Parameter:<br>TCM.DLL pCM4                             |
| Wam if outdated Executable is loaded Manage Component Vie   | Wam if outdated Executable is loaded                               |
| OK Car  | ucel Defaults Help   |

图 3.4 debug 设置

在线使用 J-LINK 调试,只需要设置 ORT:选项,如图 3.5 红框所示,调试方式选择 SW, 点击确定,保存设置。退出工程设置。

| Cortex JLink/JTrace Target Driver Setup  |  |
|--|--|
| Debug Trace Flash Download<br>J-Link / J-Trace Adapter<br>SN: 59700618<br>Device: J-Link<br>HW : V9.70 dll V6.50b<br>FW : J-Link V9 compiled May:<br>ort: Max<br>SW V<br>JTAG<br>SW Auto Clk | SW Device<br>SWDI( Ox2BAO ARM CoreSight SW-DP Up<br>Down C Automatic Detectic ID CODE:<br>C Manual Configurati Device Name:<br>Add Delete Update IR len: |
| Connect & Reset Options<br>Connect: Normal Vieset: Core  | Cache Options<br>Cache Code<br>Cache Memory  |
| Interface<br>© USB C TCP/IP<br>Scan<br>State: ready<br>TCP/IP<br>Network Set<br>IP-Addres<br>127 0   | Misc           Ting           Port         Autodetect         JLink Info           . 0 . 1         : 0         Ping         JLink Cmd                    |

图 3.5 J-LINK 设置

在进行在线调试前,需要将工程编译,并且没有错误,如图 3.6 红框所示,从左到右三 个图标的功能分别是:编译当前文件<sup>ジ</sup>、编译工程修改过的文件<sup>III</sup>、编译工程所有相关的文 件<sup>IIII</sup>。每次调试前,都需要将工程相关文件编译连接起来,并且没有错误。

| 😨 F:\SVN_other\LPC54018快速入门\SDK\boards\lpcxpre |        |      |      |                |       |       |       |       |               |      |
|--|--------|------|------|----------------|-------|-------|-------|-------|---------------|------|
| File   | Edit \ | /iew | Proj | ect            | Flash | D     | ebug  | Per   | ripher        | als  |
| 🗅  | 💕 🔒    | 9    | ž    | E <sub>B</sub> |       | 5     | 6     | -     | $\Rightarrow$ | la . |
| ۲  |        | 🧼 ·  | -    |                | a g   | pio_l | ed_ou | utput | debu          | ig 💌 |

图 3.6 编译工程

进入在线调试,在编译连接工程成功后,点击如图 3.7 红框所示 **Q**·图标即可进入调试。 在已经进入在线调试模式时,再次点击该图标,即可退出在线调试。

| utput.uvprojx - μVision |   |               |  |  |  |  |  |  |  |  |
|-------------------------|---|---------------|--|--|--|--|--|--|--|--|
| B. 🕫                    | ۰ | • · 🔗 🚓 • 🔳 🔦 |  |  |  |  |  |  |  |  |

图 3.7 进入在线调试

如图 3.8 所示为在线调试页面,图 3.8 中红框所示图标从左到右功能分别为:复位 篇、 全速运行 题、暂停 ③、步进 ④、步越 ④、步出 ④、运行至光标选中行 10、跳转至当前暂停 位置 →。

| Ψ,  | F:\SVN   | I_other∖ | LPC54018 | 快速入      | ï]\sdk\ł | ooards\lp                | схрі | esso5 | 54018\d | river_ | exam     |
|-----|----------|----------|----------|----------|----------|--------------------------|------|-------|---------|--------|----------|
| Fil | e Edit   | View     | Project  | Flash    | Debug    | Periphe                  | rals | Tools | SVCS    | Wir    | ndow     |
| ļ   | ן 🔁      |          | 8 B      | <b>B</b> | 5 6      | $\Leftarrow \Rightarrow$ | P    | 12    | 19. 19. |        | <u>ا</u> |
| RS  | <b>.</b> | . 🛞 📑    | e) 🗗 ()  | P* *()   | ۵        | 10. 📭                    |      |       | 🔊 • 🚺   | •      | <b>1</b> |
| Reg | isters   |          |          |          | џ 🔝      | Disassemb                | oly  |       |         |        |          |

图 3.8 在线调试页面

©2019 ZLG Technology Corp.,Ltd.

# 广州立功科技股份有限公司 EasyARM-LPC54018 快速入门指南 EasyARM-LPC54018 入门

在此次调试中,设置好板级硬件设置后,我们只需要在在线调试页面点击,全速运行,按下 KES/RES 按键,即可看到程序运行现象(LED1 闪烁),随后点击图 3.7 红框所示图标,退出调试。

#### 3.1.3 使用 J-Link 下载离线调试程序

在 3.1.2 中介绍的是程序在线调试的方法,实际的程序没有下载到开发板上,退出调试模 式后,开发板就不能运行调试程序了。在本小节将介绍怎么将程序下载到开发板上,在断电 上电后程序也不会丢失的调试方法。

在调试前,需要搭建一下离线调试的开发环境。将"3.集成开发环境"中的的编程算法 文件 LPC540xx\_IS25LP064A\_SPIFI.FLM 复制到 KEIL 安装目录 Keil\_v5\ARM\Flash 文件夹 下,xip\_debug.ini 文件复制到将要打开的工程路径的 MDK 文件夹下。至此基本下载环境搭 建完成。

双 击 "boards\lpcxpresso54018\driver\_examples\gpio\led\_output\mdk" 目 录 下 的 gpio\_led\_output.uvprojx 打开工程,点击如图 3.9 红框所示 图标进入工程项目管理设置。

| 惠入自 | ]\SDK\   | boar  | ds\lpc        | xpr | esso  | 5401 | 18\d | lrive | er_e | exan | nple | s\gpi |
|-----|----------|-------|---------------|-----|-------|------|------|-------|------|------|------|-------|
| ash | Debug    | Per   | iphera        | ls  | Tools | S    | VCS  | ۷     | Vin  | dow  | He   | elp   |
| 4   | 2 6      | -     | $\Rightarrow$ | P   | 帮     | 1    | R    | -     |      | *    | //≞  | //#   |
| gp  | io_led_o | utput | debu          | g 🔻 | *     | đ    | 5    | 4     | ٠    | ~    |      |       |
|     |          |       | w h           |     |       |      |      |       |      |      |      |       |

图 3.9 创建新的工程组

点击如图红框所示图标,创建一个新的工程项目选项,新建项目命名为"gpio\_led\_output qspi\_xip"的 Project Targets,点击确认,生成一组新的文件标签,并单击选择此标签,点击 "OK"保存设置,退出。

| Manage Project Items   | <b>X</b>                        |
|--|---------------------------------|
| Project Items   Folders/Extensions   Books   |                                 |
| Project Targets:<br>gpio_led_output debug<br>gpio_led_output debug<br>gpio_led_output cloude<br>cpio_led_output cloude<br>doc<br>dvces<br>device<br>utilities<br>component-saria<br>component-saria<br>component-lists<br>startup<br>CMSIS | Files: ★ ★<br>gpio_led_output.c |
| Set as Current Target  | , <u>A</u> dd Files             |
| OK Cancel  | Kelp                            |

图 3.10 创建新的工程项目

随后选择调试方式,如图 3.11 红框所示,选择刚刚创建的工程项目 gpio\_led\_output qspi\_xip,

产品入门手册



| Vi F   | SVN_    | other\L  | PC54018  | 3快退  | Ē入门\SD               | K\boa              | rds∖lp             | cxpre   | esso5 | 6401 | .8\0 |
|--------|---------|----------|----------|------|----------------------|--------------------|--------------------|---------|-------|------|------|
| File   | Edit    | View     | Project  | Fla  | sh Deb               | ug Pe              | eriphera           | als     | Tools | S    | VCS  |
|        | 📔 🖌     | 1 🥥      | X D      |      | 19 (                 | >   🔶              | -                  | M       | 鼎     | 11   | ß    |
| ٩      |         | 🖞 🗳 🗄    | - 🔍 🕴    | .0AD | gpio_le              | d_outpu            | it qspi_           | xij 👻   | ×     | 4    | 5 9  |
| Projec | t       |          |          |      | gpio_leo<br>gpio_leo | l_outpu<br>l_outpu | t debu<br>t releas | g<br>se |       | tput | .c   |
|        | \$Proje | ct: gpic | _led_out | put  | apio leo             | d outpu            | t qspi             | xip     | _     | /*   |      |

图 3.11 调试方式选择

点击如图 3.12 红框所示 图标进入工程设置页面。

| 18  | 快速入      | `]\SDK\b  | oards\lpcxp  | presso5 | 54018\dri | ver_exan | nples\gp            |
|-----|----------|-----------|--------------|---------|-----------|----------|---------------------|
| :t  | Flash    | Debug     | Peripherals  | Tools   | SVCS      | Window   | Help                |
| à   | <b>2</b> | 2 6       | ← →   🤻      | 1       | 19 B      |          | //≣_// <sub>Ř</sub> |
| L0  | 🕴 🛛 gp   | io_led_ou | tput debug [ | •       | 🛔 🗟       | 🔶 🔶      | <u>@</u>            |
|     |          | <b>џ </b> | 📄 gpio       | _led_ou | tput.c    |          |                     |
| uto | ut       |           |              | -       | · ·       |          |                     |

图 3.12 工程设置

如图 3.13 红框所示,在工程设置里,需要对 ASM 选项卡进行设置。在 Define 文本框内 填入预定义宏,"CPU\_LPC54018,\_\_CC\_ARM,DEBUG,KEIL, DEBUG, XIP\_IMAGE, W25Q128JVFM"。

| 🕅 Options for Target 'gpio_led_output qspi_xip'   |
|---|
| Device   Target   Output   Listing   User   C/C++ Asm Linker   Debug   Utilities  |
| Conditional Assembly Control Symbols           Define:         CPU_LPC54018,CC_ARM,DEBUG,KEIL, DEBUG, XIP_IMAGE, W25Q128JVFM           Undefine:  |
| Language / Code Generation  |
| Read-Qnly Position Independent       Split Load and Store Multiple         Read-Write Position Independent         Thumb Mode       Execute-only Code         No Wamings       No Auto Includes |
| Include<br>Paths  |
| Assembler -cpu Cortex-M4.fp -gapcs=interworkpd "MICROLIB SETA 1"nowam<br>control -l .\RTE\_gpio_led_output_qspi_xip string  |
| OK Cancel Defaults Help   |

图 3.13 ASM 设置

对 Linker 选项卡的设置,如图 3.14 红框所示,添加 "--predefine="-DXIP\_IMAGE""到 Misc controls 文本框末尾,注意用空格与前面的字符分隔开。

|  | EasyARM-LPC                      |
|--|----------------------------------|
| tions for Target 'gpio_led_output qspi_xi  | p' 💌                             |
| ce   Target   Output   Listing   User   C/O  | C++ Asm Linker Debug   Utilities |
| Use Memory Layout from Target Dialog   | X/O Base:                        |
| Make RW Sections Position Independent  | <u>R</u> /O Base: 0x0000000      |
| Make RO Sections Position Independent  | R/ <u>W</u> Base 0x20000000      |
| Don't Search Standard Libraries  |                                  |
| ✓ Report 'might fail' Conditions as Errors   | disable Warnings: 6314           |
| Report 'might fail' Conditions as Errors          Scatter       LPC54018_spifi_flash.scf   | disable Warnings: 6314           |
| ✓ Report 'might fail' Conditions as Errors          Scatter       LPC54018_spifi_flash.scf         File       LPC54018_spifi_flash.scf             | disable Wamings: 6314<br>▼ Edit  |
| Report 'might fail' Conditions as Errors     Scatter     File     LPC54018_spifi_flash.scf     Misc     (///devices/LPC54018/arm/keil_lit controls | disable Warnings: 6314           |

<u> 76</u>

图 3.14 Linker 选项卡

如图 3.15 所示,点击 Scatter File 下的 edit 按钮,随后点击 "OK",保存前面的修改,并 退出工程设置,将修改 sct 文件内的内容。

| 🔣 Options fo   | r Target 'gpio_led_output qspi_xip   | o'  |                                 | <b>—</b> ×- |
|--|--|---|---------------------------------|-------------|
| Device   Targ  | get   Output   Listing   User   C/C-   | ++ Asm Linker   | Debug   Utilities               |             |
| ☐ <u>U</u> se Mem<br>☐ Make<br>☐ Make<br>☐ Do <u>n</u> t<br>☞ Repo | ory Layout from Target Dialog<br>RW Sections Position Independent<br>RO Sections Position Independent<br>Search Standard Libraries<br>rt 'might fail' Conditions as Errors | <u>X</u> /O Base:<br><u>R</u> /O Base:<br>R/ <u>W</u> Base<br>disable Warnings: | 0x0000000<br>0x20000000<br>6314 |             |
| Scatter<br>File  | LPC54018_spifi_flash.scf   |   | <b>.</b>                        | Edit        |
| <u>M</u> isc<br>controls   | /////devices/LPC54018/arm/k  | :eil_lib_power.lib -remove  | -legacyalign                    | *<br>*      |
| Linker<br>control<br>string  | cpu Cortex-M4.fpscatter "./RTE/Dev<br>*.o  | rice/LPC54018JET180/LI  | PC54018_spifi_flash.scf"        | ۸<br>۳      |
|  | 0K   | Cancel De   | faults                          | Help        |

图 3.15 LPC54018\_spifi\_flash.scf 设置

| 由          |                 |            |       |                           |
|------------|-----------------|------------|-------|---------------------------|
| #if (defin | ned(XIP_IMAGE)) |            |       |                           |
| #define    | m_text_start    | 0x100001C4 |       |                           |
| #define    | m_text_size     | 0x00FFFE3C |       |                           |
| #else      |                 |            |       |                           |
| 产品)        | 、门手册            |            | ©2019 | ZLG Technology Corp.,Ltd. |



| #define    | m_text_start   | 0x000001C4 |
|------------|----------------|------------|
| #define    | m_text_size    | 0x0002FE3C |
| #endif     |                |            |
| 修          | 改成             |            |
| #if (defin | ed(XIP_IMAGE)) |            |
| #define    | m_text_start   | 0x100001C4 |
| #define    | m_text_size    | 0x007FFE3C |
| #else      |                |            |
| #define    | m_text_start   | 0x000001C4 |
| #define    | m_text_size    | 0x0002FE3C |
| #endif     |                |            |

再次进入工程设置窗口,如图所示,切换到 Debug 选项卡,设置调试器和下载器,首先 更改 Initialization File 下的 ini 文件。ini 文件选中复制进去的文件 xip\_debug.ini 点击打开。

| 👿 Options for Target 'gpi                                 | o_led_output qspi_xip'  |  | 23                 |
|---|---|--|--------------------|
| Device   Target   Output                                  | Listing   Vser   C/C++   Asm   Linker   | Debug Utilities  |                    |
| ○ Use Simulator <u>with</u><br>□ Limit Speed to Real-Time | restrictions Settings © Use: J-LINK   | ( / J-TRACE Cortex ▼                                     | Settings           |
| Load Application at Startu<br>Initialization File:        | up 🔽 Run to main() 🖾 Load Applica<br>Initialization File:<br>Edit    ilink_sramx_debu | ion at Startup 🔽 Run to r                                | main()<br>Edit     |
| Select Target Debugge                                     | er Initialization File  |  |                    |
| ≪ gpio ►  | led_output ▶ mdk ▶   ▼  | <b>∳</b> 搜索 mdk  | ٩                  |
| 组织 ▼ 新建文件夹  |   | := -   |                    |
|   | 名称  | 修改日期   | 类型                 |
| 篇库<br>册 Git<br>⊜ Subversion                               | ⊯ debug<br>⊯ output<br>⊗ jlink sramx debuq.ini  | 2019/10/16 14:31<br>2019/10/16 14:14<br>2019/10/15 15:22 | 文件夹<br>文件夹<br>配置设置 |
| ■ 视频 🗉  | 🖉 JLinkSettings.ini   | 2019/10/15 15:22   | 配置设置               |
| ■ 图片  | 🖉 xip_debug.ini   | 2019/8/29 13:57  | 配置设置               |
| <ul> <li>■ 文档</li> <li>▲) 音乐</li> </ul>                   |   |  |                    |
| 🖳 计算机   |   |  |                    |
| 🊨 本地磁盘 (C:) 🍸   | •   |  | 4                  |
| 文件  | 名(N): xip_debug.ini   | ▼ Ini Files (*.ini)<br>打开(O)                             | ▼                  |

图 3.16 调试器设置

随后进行调试器的选择和设置。本次以 J-Link 为例演示,点击选项卡中的 Settings 按钮进入 J-Link 设置界面。J-Link 设置的 Debug 界面如图 3.17 所示,调试端口选择 SW 接口。

| Cortex II ink/ITrace Target Driver Setu  |   |
|--|---|
| Debug Trace Flash Download   | ab  |
| J-Link / J-Trace Adapter<br>SN:<br>Device: J-Link<br>HW : V9.70 dll V6.50b<br>FW : J-Link V9 compiled May :<br>ort: Max<br>SY V SMHz V<br>Auto Clk | SW Device       Move         SWDII       DDCDE       Device Name         O DxOBC1       ARM CoreSight SW-DP       Up         Down       Down       Down         © Automatic Detectic       ID CODE: |
| Connect & Reset Options<br>Connect: Normal Veset: Con<br>Reset after Conn  | re Cache Options Download Options<br>Verify Code Download to Flash  |
| Interface     TCP/IP       © USB     TCP/IP       Scan     IP-Addres       State: ready     127  | Settings<br>s Port Autodetect JLink Info<br>0.0.1:0 Ping JLink Cmd  |

图 3.17 J-Link

在图 3.17 中,选中 Flash Download 选项卡,在 Download Function 功能框选上 Erase Sector, 并勾上 Program, Verify,修改编程算法的 RAM 大小为 0x8000,并添加我们之前复制到 KEIL 的 FLASH 文件夹下的编程算法文件 LPC540xx\_IS25LP064A\_SPIFI.FLM,最后确认完成设置,保 存退出。如图 3.18 所示。

| wnload Function<br>C Erase Full C<br>C Erase Sector<br>Do not Erase<br>ogramming Algorithm   | Program<br>Verify<br>Reset and Ru  | n   | lgorithm<br>20000000 ize <mark>: 0x8000</mark>   |   |
|--|--|---|--|---|
| Description  | Device Size  | Device Type   | Address Range  |   |
| Add Flash Programming Alg<br>Description<br>LPC540xx MT250L128 SPIFI   | gorithm  | Device Type<br>Fot Flash SPI  | Origin<br>Device Family Package  | × |
| LPC540xx MX25L12835FM2<br>LPC540xx W25Q128JVFM S<br>AM29x128 Flash<br>K8P5615UQA Dual Flash<br>LPC18xx/43xx MX25V8035F   | 16M<br>16M<br>16M<br>64M<br>. 8M   | Ext. Flash SPI<br>Ext. Flash SPI<br>Ext. Flash 16-bit<br>Ext. Flash 32-bit<br>Ext. Flash 32-bit<br>Ext. Flash SPI   | Device Family Package<br>Device Family Package<br>MDK Core<br>MDK Core<br>MDK Core   | ш |
| LPC540xx MX25L12835FM2<br>LPC540xx W25Q128JVFM S<br>AM25x128 Flash<br>K8P5615UQA Dual Flash<br>LPC18xx/43xx MX25V8035F<br>LPC18xx/43xx S25FL032 SP<br>LPC18xx/43xx S25FL032 SP.I.<br>LPC407x/8x S25FL032 SP.I.<br>LPC540xx IS25LP064A_SPIF<br>LPC540xx IS25LP064A_SPIF<br>LPC5460x IAP 512kB Flash<br>LPC5460x MT25QL128 SPIFI | . 16M<br>. 16M<br>. 64M<br>. 8M<br>. 8M<br>. 8M<br>. 8M<br>. 8M<br>. 8M<br>. 9M<br>. 512k<br>. 16M | Ext. Rash SPI<br>Ext. Rash SPI | Device Family Package<br>Device Family Package<br>MDK Core<br>MDK Core | = |

产品入门手册

©2019 ZLG Technology Corp.,Ltd.



#### 图 3.18 FLASH

至此,软件设置基本完成,需要将工程编译,并且没有错误,如图 3.6 红框所示,从左 到右三个图标的功能分别是:编译当前文件、编译工程修改过的文件、编译工程所有相关的 文件。在下载前,需要依次点击三个图标,也可以直接点击"编译工程所有相关的文件",等 待编译通过后,点击如图 3.19 红框所示题图标,将程序下载到开发板上,复位开发板,按下 KES/RES 按键,即可看到程序运行现象(LED1闪烁)。



图 3.19 下载



# 4. 免责声明

广州周立功单片机科技有限公司随附提供的软件或文档资料旨在提供给您(本公司的客户)使用,仅限于且只能在本公司制造或销售的产品上使用。

该软件或文档资料为本公司和/或其供应商所有,并受适用的版权法保护,版权所有。如 有违反,将面临相关适用法律的刑事制裁,并承担违背此许可的条款和条件的民事责任。

本公司保留在不通知读者的情况下,有修改文档或软件相关内容的权利,对于使用中所 出现的任何效果,本公司不承担任何责任。

该软件或文档资料"按现状"提供,不提供保证,无论是明示的、暗示的还是法定的保证。这些保证包括(但不限于)对出于某一特定目的应用此软件的适销性和适用性默示的保证。 在任何情况下,公司不会对任何原因造成的特别的、偶然的或间接的损害负责。



# 销售与服务网络

#### 广州周立功单片机科技有限公司

地址:广州市天河区龙怡路 117 号银汇大厦 16 楼 邮编: 510630 电话: 020-38730916 38730917 38730976 38730977 网址: www.zlgmcu.com 传真: 020-38730925

#### 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 传真: 020-87578842



#### 南京周立功

地址:南京市秦淮区汉中路 27 号友谊广场 17 层 F、 GΣ 电话: 025-68123901/68123902/68123919 传真: 025-68123900

#### 重庆周立功

地址:北京市海淀区紫金数码园 3 号楼(东华合创大 地址:重庆市渝北区龙溪街道新溉大道 18 号山顶国 宾城 11 幢 4-14 电话: 023-68796438/68796439/68797619 传真: 023-68796439

#### 成都周立功

地址:成都市一环路南二段1号数码科技大厦403 室 电话: 028-85439836/85432683/85437446 传真: 028-68796439

深圳周立功(二部) 1705 电话: 0755-83781788/83782922 传真: 0755-83793285

#### 上海周立功

地址:上海市黄浦区北京东路 668 号科技京城东座 12E 室 电话: 021-53083451/53083452/53083453 传真: 021-53083491

#### 北京周立功

厦) 8 层 0802 室 电话: 010-62635033/62635573/62635884 传真: 010-82164433

#### 杭州周立功

地址: 杭州市西湖区紫荆花路 2 号杭州联合大厦 A 座4单元 508 电话: 0571-89719484/89719499/89719498 传真: 0571-89719494

#### 深圳周立功 (一部)

地址:深圳市福田区深南中路 2072 号电子大厦 1203 地址:深圳市坪山区比亚迪路大万文化广场 A 座 室 电话: (0755)82941683/82907445 传真: (0755) 83793285

#### 武汉周立功

地址: 武汉市武昌区武珞路 282 号思特大厦 807 室 电话: 027-87168497/87168297/87168397 传真: 027-87163755

#### EasyARM-LPC54018 快速入门指 州立功科技股份有限公司

EasyARM-LPC54018 入门

周立功厦门办 地址: 厦门市思明区厦禾路 855 号英才商厦 618 室 电话: 18650195588

#### 周立功苏州办

地址: 江苏省苏州市广济南路 258 号 (百脑汇科技中 心 1301 室) 电话: 0512-68266786 & 18616749830

地址:浙江省宁波市高新区星海南路 16 号轿辰大厦

#### 周立功合肥办

地址: 安徽省合肥市蜀山区黄山路 665 号汇峰大厦 1607 电话: 13851513746

#### 周立功天津办

地址:天津市河东区十一经路与津塘公路交口鼎泰大 地址:山东省青岛市李沧区青山路 689 号 宝龙公寓 厦 1004 室 电话: 18622359231

#### 周立功山东办

电话: 0574-87228513 / 87229313

周立功宁波办

1003

3 号楼 311 电话: 13810794370

#### 周立功郑州办

地址:河南郑州市中原区百花路与建设路东南角锦绣 地址: 沈阳市浑南新区营盘西街 17 号万达广场 A4 华庭 A 座 1502 室 电话: 17737307206

#### 香港周立功

地址:香港新界沙田火炭禾香街 9-15 力坚工业大厦 13 层 电话:(852)26568073 26568077

#### 周立功沈阳办

座 2722 室 电话: 18940293816

#### 周立功长沙办

地址:湖南省长沙市岳麓区奥克斯广场国际公寓 A 栋 2309 房 电话: 0731-85161853