ACM32F0x0/FP0X 系列芯片开发快速上手

目录

1.	开发	:板资源概述	2
	1.1.	开发板板载资源	2
2.	板载	硬件介绍	5
	2.1.	MCU之 ACM32F070	5
	2.2.	CMSIS-DAP 调试下载接口/虚拟 USB 转串口	6
	2.3.	板载 LED	6
	2.4.	按键和 BOOT 跳帽	7
	2.5.	电源输入输出	8
	2.6.	板载时钟	9
	2.7.	IO 口	9
3.	开发	环境搭建	.11
	3.1.	设备驱动安装	.11
	3.2.	仿真环境搭建	.13
		3.2.1. MCU 选择	.13
		3.2.2. 下载口选择	.14
		3.2.3. 下载算法选择	.15
		3.2.4. 调试模式配置	.16
		3.2.5. BOOT 引脚配置	.17
	3.3.	KEIL 工程移植	.18
	2.01		-0

1. 开发板资源概述

1.1.开发板板载资源

- MCU: ACM32F070, LQFP64(10mm*10mm), FLASH: 128KB, SRAM: 32KB;
- 1个电源指示灯,1个Link指示灯,1个用户指示灯。
- 1个电源供应/接入口。
- 1个启动模式选择跳帽,选择芯片启动模式。
- 1个系统复位按钮,用于复位MCU。
- 1个用户功能按钮。
- 提供了CMSIS-DAP方式下载、调试,USB虚拟串口打印功能。
- 所有IO口全部引出,包括晶振占用的IO口。
- 开发板支持外接14个触摸按键和段码式显示屏,开发包中有原理图。

AisinoChip



图 1-1 ACM32F070 Layout



图 1-2 ACM32F070 实物图

Copyright © 2022 上海爱信诺航芯电子科技有限公司

2. 板载硬件介绍

2.1.MCU之 ACM32F070



2.2. CMSIS-DAP 调试下载接口/虚拟 USB 转串口

采用 ACH512 芯片作为下载调试/串口打印芯片,当用户通过 USB 线将开发板与 PC 机连接 后,调试/下载程序时在 KEIL 中选择 CMSIS-DAP Debugger 模式,并且在设备管理器中可以找到 航芯虚拟串口端口。(后续会具体介绍开发环境搭建)



2.3. 板载 LED

用户程序灯(USR),绿色,辅助用户查询/调试程序。 电源指示灯(PWR),绿色,MCU上电时亮,下电时熄灭。 Link指示灯(Link),红色,反映芯片的下载/调试状态。



图 2-4 板载 LED

2.4. 按键和 BOOT 跳帽

MCU 复位按键(RstnPB)红色,用于复位主控芯片,按下按键芯片复位。

用户按键(UserPB)黑色,人机交互按键。

芯片启动模式选择跳帽,连接跳帽,芯片会运行用户代码(eFlash);断开跳帽,芯片只会运行自有 Boot 程序(ROM 中)。



图 2-4 用户按键



2.5. 电源输入输出

通常使用 USB 供电,板载 LDO 将 5V(USB_Vbus)转成 3.3V, JP5 默认用跳帽短接。当不使用 USB 供电时,可以断开 JP5,通过排针从外部接入电源给板载 MCU 供电。MCU 的 VDD 输入电 压范围是 1.7V~3.6V。



2.6. 板载时钟

板上默认有外部高速 8MHz 无源晶振,外部低速 32.768KHz 无源晶振。程序可以使用片内 的 RC64M 时钟作为系统时钟,也可以使用 8MHz 的外部晶振加 PLL 输出的时钟作为系统时钟。 32.768KHz 的外部晶振时钟可用作 RTC 的工作时钟。



图 2-8 片外时钟

2.7.IO 口

芯片所有 IO 引出到排针 CN1、CN2、CN3 上。并且 JP3、JP4、JP5、JP6 上都提供电源/地, Copyright © 2022 上海爱信诺航芯电子科技有限公司 9 / 24

可以灵活给接插的扩展子板输入/输出电源。



图 2-9 扩展 IO 接口

3. 开发环境搭建

3.1.设备驱动安装

用户第一次将开发板与 PC 机的 USB 端口相连, Win7 系统用户需要安装 Link 芯片的驱动 (Win10 系统用户不需要)。

🔺 🔓 其他设备

AISINOCHIP CMSIS-DAP

选择此设备,右键更新驱动,直接点下面的->从驱动列表中选择(COM 类)

0	① 更新驱动程序软件 - AISINOCHIP CMSIS-DAP	
	浏览计算机上的驱动程序文件	
	在以下位置搜索驱动程序软件: <pre>chip release\ACM32F030开发包-20200624\release\USBDriver ▼ 浏览(R)</pre> ☑ 包括子文件夹(I)	
	→ 从计算机的设备驱动程序列表中选择(L) 此列表将显示与该设备兼容的已安装的驱动程序软件,以及与该设备处于同一类别下的 所有驱动程序软件。	
	下一步(N) 取消	
	图 3-1 选择驱动程序	

Y

从以下	列表选择设备的	类型。					
常见硬件	₩ <u>类型(</u> H):						
冒存儲	卷卷影副本					*	
◆存储	控制器						
一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	ን መደረጉ እስከ 1980 እስከ 1880 እስከ 1980 እስከ 1880 እስከ 1						
♥端] (COM 和 LPT)						
龙萝叶							
<u></u> () () () () () () () () () ()	前能适配器					_	
● 3FR 『『 4TA	」插即用巡动程序 k线设备					_	
■ ↓1算	I 现						
▶ 监视	144						
一键盘						-	
简唱	「用神器に音						
					下一步(N)	取消	
							J
		图	3-2 选择	驱动程序			
		冬	3-2 选择	驱动程序			
更新	驱动程序软件 - AISIN	S OCHIP CI	3-2 选择 ^I	驱动程序			
更新	驱动程序软件 - AISIN	冬 ОСНІР СІ	3-2 选择耳	驱动程序			
』 更新	驱动程序软件 - AISIN 更为此硬件安装的	图 OCHIP CI	3-2 选择 MSIS-DAP 动程序	驱动程序			
更新 选择要	驱动程序软件 - AISIN 医为此硬件安装的 语注注:###################################	图 OCHIP CI 设备驱动	3-2 选择 MSIS-DAP 动程序	驱动程序	左旬今更少选约	ur chife céoluité	
① 更新 选择要	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 , 然后单击"下一	驱动程序 步",如果手头	有包含要安装的	驱动程序的磁	
』 更新 选择要	驱动程序软件 - AISIN 为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 , 然后单击"下一	驱动程序 ^{步",如果手头}	有包含要安装的	驱动程序的磁	
』 更新 选择要	驱动程序软件 - AISIN 医为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 , 然后单击"下一	驱动程序 步"。如果手头	有包含要安装的	驱动程序的磁	
 ■ 更新 选择男 	驱动程序软件 - AISIN 医为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 , 然后单击"下一	抠动程序 ^{步",如果手头}	有包含要安装的	驱动程序的磁	
』 更新 选择要 ✓	驱动程序软件 - AISIN 医为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘	图 OCHIP CC 设备驱动 商和型导 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号	驱动程序 步",如果手头	有包含要安装的	驱动程序的磁	
』 更新 选择要 ✓ 厂商 (际職)	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和关型)	图 OCHIP CI 设备驱范 奇和型号 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号	驱动程序 步"。如果手头 □	有包含要安装的	驱动程序的磁	
』更新 选择要 ≪ 「商 (标准録 Broth Comp	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和J类型) er ag GSM Badio Card	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号 氯ECP 打印机满口 习名口通信端口	驱动程序 步"。如果手头	有包含要安装的	驱动程序的磁	
』更新 选择要 ≪ 「商 (标態 Broth Comp DBC	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和J类型) er aq GSM Radio Card	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号 氯ECP 打印机端 氯打印机端口 氯多口通信端口 到通信端口	驱动程序 步"。如果手头 □	有包含要安装的	驱动程序的磁	
』更新 选择要 「商 (标電 Broth Comp DBC	驱动程序软件 - AISIN 要为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和 和 aq GSM Radio Card III	图 OCHIP CC 设备犯还 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号 国ECP 打印机满 国子印通信满口 副多口通信满口 副通信满口	驱动程序 步",如果手头	有包含要安装的	驱动程序的磁	
』更新 选择理	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和J类型) er aq GSM Radio Card III	图 OCHIP CI 设备驱范 奇和型号 安装"。	3-2 选择 MSIS-DAP 动程序 , 然后单击"下一 型号 副ECP 打印机端 副方印通信端口 副通信端口	驱动程序 步"。如果手头	·有包含要安装的	驱动程序的磁 曲安装(H)	
』更新 选择要 「商 「际 Broth Comp DBC ↓	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和 和 和 和 和 和 和 和 和 和 和 和 和	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号 氯ECP 打印机端口 氯多口通信端口 氯通信端口	驱动程序 步"。如果手头	有包含要安装的	驱动程序的磁 盘安装(H)	
』更新 选择要 ✓ 「商 (标時 Brothe Comp DBC <	驱动程序软件 - AISIN 更为此硬件安装的 请选定硬件设备的厂 盘,请单击"从磁盘 和J类型) er aq GSM Radio Card III 个驱动程序已经过数字: 拆我为什么驱动程序签	图 OCHIP CI 设备驱动 商和型号 安装"。	3-2 选择 MSIS-DAP 动程序 ,然后单击"下一 型号 国ECP 打印机端 可印机端口 副多口通信端口	驱动程序 步",如果手头 □	有包含要安装的	驱动程序的磁 盘安装(H)	

	X	更多操作	
◎ 更新驱动程序软件 - AISINOCHIP CMSIS-DAP	🌆 查找文件		
生权再为此项件它准的沿发取动程序	查找范围(I):	퉬 USBDriver	- G 🜶 📂 🖽 -
20年安乃此硬件安泰的反雷驱动性力	9	名称	修改日期 类型
请选定硬件设备的厂商和型号,然后单击"下一步"。如果手头有包含要安装的驱动 ————————————————————————————————————	最近访问的位置	AisinoChipCDC.inf	2020/6/18 9:22 安装信息
从磁盘安装			
插入割盖商的安装盘,然后确定已在下面选定正 确定 确定 通问驱动器。	桌面	选择固	件包中的
取消		AisinoChipC	DC.inf
) 雨 (赤准護	库		
Brother			
Compa 利益两义计复制米源 U/: DBC D:\Aisinochip release\ACM32F030开发包-20 ▼ 浏览 (8)	计算机		
		<	
□ 这个驱动程序已经过数字签名。 从磁盘安	网络	文件名(W): AisinoChipCDC.inf	▼ 打开 (0)
告诉我为什么驱动程序签名很重要		文件类型(I): 安装信息 (*.inf)	- 取消

图 3-4 选择驱动程序

忽略警告提示,继续安装,安装完毕。后续调试程序可以选择此 COM 口获得打印信息。

● ① 更新驱动程序软件 - AisinoChip Virtual Com Port (COM20)	
Windows 已经成功地更新驱动程序文件	
Windows 已经完成安装此设备的驱动程序软件:	
AisinoChip Virtual Com Port	
图 3-5 选择驱动程序	

3.2. 仿真环境搭建

3.2.1. MCU 选择

安装航芯开发包里的 pack 包"Aisinochip.ACM32F0X0.1.0.0.pack",安装完成,选择所使用的芯片 具体型号,如图所示:

evice Target Output Listing User Software Packs	C/C++ (ACG) Asm Linker Debug Vtilities	
Vendor: Aisinochip	Software Pack	_
Device: ACM32F030R8T7	Pack: Aisinochip.ACM32F0X0.1.0.0	
Toolset: ARM	URL: www.keil.com/dd2	
Search:		
Aisinochip AcM32F0X0 Series ACM32F030 ACM32F030C8T7 ACM32F030C8T7 ACM32F030K8T7 ACM32F030K8U7 ACM32F030K	The ACM32F0X0 device family contains an ARM Cortex-M0 processor, running up to 64 MHz with a versatile set of on-chip peripherals.	~
ОК	Cancel Defaults He	lp

3.2.2. 下载口选择

连接调试器:如图所示,切换到 Debug 页面后在下拉框中选择 CMSIS-DAP Debugger,然后 点击 Settings 按钮,显示图 3-7 的调试器连接情况。

🕅 Options for Target 'ACM32F030'	×
Device Target Output Listing User C/C++ A	sm Linker Debug Utilities
C Use Simulator <u>with restrictions</u> <u>Settings</u> ☐ Limit Speed to Real-Time	
✓ Load Application at Startup Run to main() Initialization File: ✓ debug.ini Restore Debug Session Settings ✓ Breakpoints ✓ Toolbox ✓ Watch Windows & Performance Analyzer ✓ Memory Display ✓ System Viewer CPU DLL: Parameter: SARMCM3.DLL	✓ Load Application at Startup Run to main() Initialization File: Restore Debug Session Settings ✓ Breakpoints ✓ Toolbox ✓ Watch Windows ✓ Memory Display ✓ System Viewer Driver DLL: Parameter: SARMCM3.DLL
	wer Description nies
OK Can	cel Defaults Help
图 3-7 调试 CMSIS-DAP Cortex-M Target Driver Setup Debug Trace Flash Download	器类型 ×
CMSIS-DAP - JTAG/SW Adapter SW Device	
CMSIS-DAP AISINOCHIP V1 IDCODE Serial No: CMSIS-DAP AISIN Firmware Version: 1.10 Image: SWJ Port: SWDIO Max Clock: 5MHz	
Debug Connect & Reset Options Connect: Normal Reset: Autodetect Reset after Connect Log Debug Accesses Stop after Reset	Cache Options Cache Code Cache Code Cache Memory
ОК	Cancel Help
 OK Can	cel Defaults Help

图 3-8 调试器连接情况

3.2.3. 下载算法选择

3.2.1 中的 pack 包包含了算法文件,安装完后会自动将算法文件拷贝到正确的路径。也可

以手动操作,将开发包中的 ACM32F0x0_eflash.flm 拷贝到 C:\KEIL\ARM\Flash 目录下。然后在图 的基础上点击 "Flash Download" 按钮进入插件选择界面,选择烧录插件。

Download Function C Erase Full Chip C Erase Sectors C Do not Erase	 ✓ Program ✓ Verify ☐ Reset and F 	RAM for / Start:	Algorithm 0x20000000	Size: 0x00001000	
Description	Device Size	Device Type	Addre	ess Range	
ACM32F0x0 128KB Flash	128k	On-chip Flash	00000000	H - 0001FFFFH	
ACM32F0x0 128KB Flash	128k	On-chip Flash	00000000	H - 0001FFFFH	_

图 3-9 添加烧录插件

3.2.4. 调试模式配置

按如图所示的配置,就可以在调试前先下载程序到 eFlash 中然后开始调试程序。

une l'um Ser l'	output Listing osei	O'C' Asim Littler Bebug
Configure Flash N	Menu Command	
Use Target [Driver for Flash Programming	Vse Debug Driver
	Use Debug Driver	Settings IV Update Target before Debugging
Init File:		Edit
C Use External	Tool for Flash Programming	
Command:		
J		
Arguments:		
Arguments:	Run Independent	
Arguments:	Run Independent	
Arguments:	Error Endependent	Add Output File to Group:
Arguments:	Run Independent	Add Output File to Group:
Arguments:	Ele Processing (FCARM):	Add Output File to Group:
Arguments:	Run Independent File Processing (FCARM):	Add Output File to Group:

图 3-10 调试前先下载程序

3.2.5. BOOT 引脚配置

ACM32F0x0/FP0X两种启动模式: ROM 启动和 eFlash 启动。系统上电时,芯片会读取 安全序列字段和系统寄存器 WMR 的 BootMode 标志位,决定是将 eFlash 还是将 ROM 映射到 0x0 起始逻辑地址。BootMode 标志位由上电时 BOOT 引脚(PF3)的高低电平决定。 图描述了芯片启动模式选择过程。



图 3-11 航芯启动模式选择

故需要在不写入安全序列的情况下,将 BOOT 引脚拉低,使 ACM32F0x0/FP0X 从 eflash 启动。

3.3. KEIL 工程移植

本章节描述如何从 STM32F103 的 UART Keil 工程移植到航芯的 ACM32F0/FP0 芯片上。对比 一下 STM32F103 和 ACM32F0/FP0 工程的异同:



我们看看 STM32F103 的 UART 工程和航芯工程的对应关系以及移植方法。

1)、STM32F103 的 Drivers/HAL_Driver 对应航芯工程的 CoreDriver/HAL, 里面放着的是芯片 厂商提供的外设驱动库。

2)、STM32F103 中 CMSIS 对应航芯工程 CoreDriver/CMSIS 的 core_cm0.h 服务,此文件是 ARM 官方提供的。

3)、STM32F103 中 Example/MDK 下的.s 启动文件对应航芯工程中 CoreDriver/Device 中的.s 文件,主要是中断向量表。STM32F103 中的 system_stm32f1xx.c 对应的是航芯工程中的 System_ACM32F0x0.c, 此文件是和芯片系统相关的, System_ACM32F0x0.c 中主要是时钟和复位

功能。

4)、STM32F103 中的 User 下的 main.c 对应航芯工程中的 SourceCode/main.c 和 SourceCode/APP.c, 是应用层的逻辑, APP.c 实现的是串口的收发,将收到的数据全部发回。

5)、STM32F103 中的 stm32f10x_it.c 对应 ACM32F0x0 中 SourceCode/APP.c 中的 UART1 IRQHandler,是中断处理函数。

如上所述, 航芯的 demo 分层清晰:应用层、硬件抽象层、系统层、ARM 公共文件。应用 层的 main.c 文件中主要是系统时钟初始化, systick 初始化, 串口打印初始;而 APP.c 是 uart demo 应用层的代码,是业务相关、应用相关的逻辑处理。HAL 下的文件是硬件抽象层,如 I2C/SPI/UART 等的 HAL 文件都包含了外设初始化和通信服务,所有和外设通信、数据交互的服 务都在 HAL_xxx.c 文件中提供好了。系统层是芯片系统相关的代码,如时钟和复位。ARM 公共 文件则是处理器内核相关的代码,是 ARM 提供的标准化接口,如 NVIC 接口, systick 初始化等。

串口 Demo 移植介绍:

打开 ACM32F0/FP0 UART demo 中的 MDK_Project 文件夹下的工程文件。

1) MCU 选择, Device 选择对应的 ACM32F0/FP0。

2)如图 3-12 所示, IROM1 起始地址填 0, 是 eflash 的起始地址,大小根据产品型号的参数填写(见数据手册),最大为 128KB,开发板上 F070R8T7 芯片为 128KB 即 0x20000。IRAM1 起始地址填 0x20000000,为 SRAM 的起始地址,大小根据产品型号的参数填写。编译器选择 version 5。

3) 如图 3-13 所示,可以生产.bin 文件。

4)如图 3-14 所示,将 One ELF Section Per Function 的√打上,链接时未被调用到的函数 不会被链接进去,能大大减少生成镜像文件的大小。编译优化选项见图 3-15,链接配置见图 3-14。

AisinoChip

Options for Target 'Project'			×
Device Target Output Listing	g User C/C++ A	sm Linker Debug Utilit	ies
ARM ARMCM0	il (MHz): 12.0	Code Generation ARM Compiler: Use default co	ompiler version 5
Operating system: None	•	Use Cross-Module Optimization	n
System Viewer File:		Use MicroLIB	Big Endian
Use Custom File			
Read/Only Memory Areas		Read/Write Memory Areas	
default off-chip Start	Size Startup	default off-chip Start	Size NoInit
ROM1:	0	RAM1:	
ROM2:	0	RAM2:	
ROM3:	с	RAM3:	
on-chip		on-chip	
IROM1: 0x0 0	×10000 💿	IRAM1: 0x20000000	0x4000
IROM2:	0	IRAM2:	
	OK Cano	el Defaults	Help
	图 3-12 Ta	arget 配置	
Options for Target 'Project'			×
Device Target Output Listin	ng Vser C/C++ A	sm Linker Debug Utiliti	22
Command Items	User Command		Stop on Exi S
Before Compile C/C++ File			
Run #1		i 🔁 i	Not Specified
Null #2	1		VOL SDECITED I

Command Items	User Command		Stop on Exi	S
Before Compile C/C++ File				
Run #1		2	Not Specified	
🗌 🗌 Run #2		2	Not Specified	
Before Build/Rebuild				
Run #1		2	Not Specified	
Run #2		2	Not Specified	
After Build/Rebuild				
🔽 Run #1	fromelf.exebinoutput ./Out_Files/Project.bi	2	Not Specified	
Run #2		2	Not Specified	\Box
Run 'After-Build' Conditionally Beep When Complete	T Start Debugging			

图 3-13 User 页面

Preprocessor Symbols Define: Undefine: Language / Code Generation Execute-only Code Strict ANSI C Warnings: All Warnings Implementation Execute-only Code Strict ANSI C Warnings: All Warnings Implementation: Level 1 (-01) Enum Container always int Thumb Mode Optimize for Time Plain Char is Signed No Auto Includes Still Load and Store Multiple Baad/Optiv Position Independent
Define: Undefine: Language / Code Generation Execute-only Code Strict ANSI C Wamings: All Wamings Optimization: Level 1 (O1) Enum Container always int Thumb Mode Optimize for Time Plain Char is Signed No Auto Includes Selit Load and Store Multiple
Undefine: Language / Code Generation Execute-only Code Strict ANSI C Warnings: All Warnings Optimization: Level 1 (O1) Enum Container always int Thumb Mode Optimize for Time Plain Char is Signed No Auto Includes Setit Load and Store Multiple Bead-Only Position Independent C 99 Mode
Language / Code Generation Execute only Code Strict ANSI C Mamings: All Warnings Coptimization: Level 1 (-01) Enum Container always int Detimize for Time Plain Char is Signed No Auto Includes Solit Load and Stare Multiple Copy Mode
Execute-only Code Generation Execute-only Code Strict ANSI C Warnings: All Warnings Optimization: Level 1 (-01) Enum Container always int Thumb Mode Optimize for Time Plain Char is Signed No Auto Includes Solit Load and Store Multiple Bead-Only Position Independent C 99 Mode
Optimization: Level 1 (-0.1) Enum Container always int Thumb Mode Optimize for Time Plain Char is Signed No Auto Includes Solit Load and Store Multiple Bead-Opty Position Independent If C99 Mode
Optimize for Time In Plain Char is Signed No Auto Includes Solit Load and Store Multiple Bead Only Position Independent If C99 Mode
Solit Load and Store Multiple Bead Only Position Independent G9 Mode
✓ One <u>ELF</u> Section per Function
Include Paths Misc Controls Compiler control string
OK Cancel Defaults Help
图 3-14 编译链接选项

Options for Target 'Project'		x
Device Target Output Listing Vser C/C++	Asm Linker Debug Vtilities	
☑ Use Memory Layout from Target Dialog	X/O Base:	
Make RW Sections Position Independent	R/O Base: 0x0000000	
Make RO Sections Position Independent	B/W Base 0x2000000	
Don't Search Standard Libraries		_
Report 'might fail' Conditions as Errors	disable Warnings:	

图 3-15 链接加载

5),代码层面上,保持 CoreDriver/CMSIS、CoreDriver/Device、CoreDriver/HAL 不变,将 main 文件进行替换。系统部分的替换内容如图 3-16 所示,串口部分的替换如图 3-17 和 3-18 所示,高度相似,都是通过 HAL 接口完成。头文件使用 APP.h 替换 main.h。去除图 3-19 的内容。 去除 PUTCHAR_PROTOTYPE 以及之后的所有内容。

HAL_Init();					
/* Configure	the system	clock to	64	MHz	*/

替换成ACM32F0/FP0的System_Init(); 系统时钟配置成64M, PCLK配置为64M

图 3-16 系统配置修改

AisinoChip

```
/* Put the USART peripheral in the Asynchronous mode (UART Mode) */
  /* UART configured as follows:
     - Word Length = 8 Bits (7 data bit + 1 parity bit) : BE CAREFUL : Program 7 c
      - Stop Bit = One Stop bit
                = ODD parity

    Parity

                = 9600 baud

    BaudRate

       Marduara flow control disabled (DTS and CTS signals)
                                                    */
  UartHandle.Instance
                        = USARTx;
  UartHandle.Init.BaudRate = 9600;
  UartHandle.Init.WordLength = UART WORDLENGTH 8B;
  UartHandle.Init.StopBits = UART STOPBITS 1;
                        = UART PARITY ODD;
  UartHandle.Init.Parity
  UartHandle.Init.HwFlowCtl = UART HWCONTROL NONE;
  UartHandle.Init.Mode = UART MODE TX RX;
  if (HAL UART Init(&UartHandle) != HAL OK)
  -{
    /* Initialization Error */
    Error Handler();
  1
                         图 3-17 STM32F103 串口初始化
#define UART BAUD RATE 115200
UART HandleTypeDef UART1 Handle;
* function : Uart_Init
* Description: Uart Initiation.
void Uart_Init(void)
Ł
   UART1 Handle.Instance
                                  = UART1;
   UART1 Handle.Init.BaudRate = UART BAUD RATE;
   UART1 Handle.Init.WordLength = UART WORDLENGTH 8B;
   UART1 Handle.Init.StopBits = UART STOPBITS 1;
   UART1_Handle.Init.Parity = UART_PARITY_NONE;
UART1_Handle.Init.Mode = UART_MODE_TX_RX_DEBUG;
   UART1 Handle.Init.HwFlowCt1 = UART HWCONTROL NONE;
  HAL_UART_Init(&UART1_Handle);
ŀ
                       图 3-18 ACM32F0/FP0 的串口初始化
```

AisinoChip

```
/* Private typedef -----*/
/* Private define -----*/
/* Private macro -----*/
/* Private variables -----*/
/* UART handler declaration */
UART_HandleTypeDef UartHandle;
/* Private function prototypes -----*/
#ifdef __GNUC__
/* With GCC, small printf (option LD Linker->Libraries->Small printf
  set to 'Yes') calls __io_putchar() */
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
L#endif /* __GNUC__ */
void SystemClock Config(void);
static void Error_Handler(void);
/* Private functions -----
                                             ----*/
```

图 3-19 STM32F103 去除的部分

联系我们

公司:上海爱信诺航芯电子科技有限公司 地址:上海市闵行区合川路 2570 号科技绿洲三期 2 号楼 702 室 邮编: 200241 电话: +86-21-6125 9080 传真: +86-21-6125 9080-830 Email: <u>Service@AisinoChip.com</u> Website: <u>www.aisinochip.com</u>

版本维护

版本	日期	作者	描述
V0.1	2021-01-07	Aisinochip	初始版
V0.2	2021-02-19	Aisinochip	核心开发板芯片为 F070R8T7
V0.3	2021-03-19	Aisinochip	增加KEIL工程移植章节
V0.4	2021-03-26	Aisinochip	STM32F103到航芯芯片的HAL移植
V0.5	2022-02-04	Aisinochip	以V2开发板为基础,更新本文档

本文档的所有部分,其著作产权归上海爱信诺航芯电子科技有限公司(简称航芯公司)所有, 未经航芯公司授权许可,任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本 文档没有任何形式的担保、立场表达或其他暗示,若有任何因本文档或其中提及的产品所有资 讯所引起的直接或间接损失,航芯公司及所属员工恕不为其担保任何责任。除此以外,本文档 所提到的产品规格及资讯仅供参考,内容亦会随时更新,恕不另行通知。