

RTL8720D 开发环境搭建

1.	环境搭建注意事项	2
2.	编译环境搭建	2
	2.1 安装 Cygwin(Cygwin 和 unbunt 环境选一个即可)	2
	2.2 安装 ubuntu 环境(Cygwin 和 unbunt 环境选一个即可)	3
	2.3 拷贝代码	3
	2.4 编译	3
	2.4.1 编译 km0	3
3.	串口下载	4
4.	测试	8
5. J	J-link 下载	9
6. 5	SDK 目录结构简介	. 11

修改记录

类型	修改内容	修改人	日期	软件版本
Α	初版	杨宾	2019/06/04	-
М	增加硬件连接和代码结构简要说明	杨宾	2019/12/12	-
山と正山				

类型: A-新增 M-修改 D-删除

1. 环境搭建注意事项

B&T 博安浦

Windows 使用 Cygwin 环境, 必须使用 Cygwin 32 位版本(64 位 windows 也要使用 32 位的 Cygwin)

本教程开发环境使用的是 32 位的 Cygwin。

2. 编译环境搭建

编译环境主要有两种,一种是 window 下的 Cygwin 或者使用 ubuntu (测试环境使用的 ubuntu1604)环境搭建 (Cygwin 搭建后实际上就是一个 linux 环境,后续操作和 ubuntu 操作 基本一致)

2.1 安装 Cygwin(Cygwin 和 unbunt 环境选一个即可)

Select packages to install									
w Full V Search make Cle	ar								
ackage	Current	New		Bin?	Src? Categori	es			
/indowMaker		Skip	•		×11				
/indowMaker-debuginfo		Skip	-		Debug				
utomake		Skip	-		Devel				
utomake1.10		Skip	-		Devel				
utomake1.11		Skip	-		Devel				
utomake1.12		Skip	-		Devel				
utomake1.13		Skip	*		Devel				
utomake1.14		Skip	•		Devel				
utomake1.15		Skip	-		Devel				
utomake 1.4		Skip	-		Devel				
utomake1.5		Skip	-	H	Devel				
utomake 1.6		Skip	-	H	Devel				
tomake I. /		Skip	-	H	Devel				
tomake I.o		Skip	-	H	Devel				
aunake 1.5		Skip	-	H	Devel				
nake nako dabuginfa		Skip	-	H	Dever				
nake dep		Skip	-	H	Devel				
nake-qui		Skip	-	H	Devel				
nacs-cmake		Skin	-	H	Editors				
tra-cmake-modules		Skin	-	H	Devel				
c-tools-epoch1-automake		Skip	-	H	Devel				
cc-tools-epoch2-automake		Skip	-	П	Devel				
cmakedep		Skip	-		Devel				
ake		Skip	-		Devel				
ake-debuginfo		Skip	+		Debug				
WMaker-devel		Skip	-		Libs				
WMaker1		Skip	•		Libs				
pagemakertools		Skip	•		Graphics				
pagemaker0.0-debuginfo		Skip	-		Debug				
pagemaker0.0-devel		Skip	*		Libs				
pagemaker0.0-doc		Skip	•		Libs				
pagemaker0.0_0		Skip			Libs				
ake	4.2.1-2	Кеер	-		Devel				
Cygwin Setup - Select Packa	iges	5km	•1		I I Debug				
Select Packages Select packages to install									
<u>v</u> iew Full <u>S</u> ear	ch bc	<u>C</u> lear					 		
1.519/20			Current	3		New	Bin?	Src?	Catego
Package			Current			and a lot of the second of the second		0.0.	ouroge
Package						2			- Contrage

版本: B2 保存期限: 5年 生效日期: 2014 年 7 月 4 日 第 2 页 共 12 页



Cygwin Setup - Select Packages					
Select Packages Select packages to install					
View Full ~ Search neurses Qle	ar				
Package	Current	New		Bin?	Src?
gambas3-gb-ncurses		Skip	*		
libncurses++w10		Skip	-	6	
libncurses-devel		Skip	•		
libncursesw10	6.0-12.20171125	Кеер	•		
mingw64-i686-ncurses		Skip	•		
mingw64-x86_64-ncurses		Skip	-		
ncurses	6.0-12.20171125	Кеер	-		
ncurses-debuginfo		Skip	•		

2.2 安装 ubuntu 环境(Cygwin 和 unbunt 环境选一个即可)

- (1) 安装 ubuntu 或者虚拟机 (详细步骤可以参考网上教程), 这里使用的是 ubuntu1604
- (2) Ubuntu 命令行执行下面的指令安装依赖包

sudo apt-get install libc6-i386 lib32ncurses5 make bc gawk ncurses

2.3 拷贝代码

拷贝并解压 sdk 代码

2.4 编译

2.4.1 编译 km0

进入 sdk/project/realtek_amebaD_cm0_gcc_verification 执行 make all 编译成功结果如下

.ram_image2.nocad			
.ram_heap.data		3 551424	
.xip_image2.text		2 201326624	
.ram_retention.er		786432	
.ram_retention.te		0 786440	
.debug_info			
.debug_abbrev	48718		
.debug_loc	86908		
.debug_aranges	6608		
.debug_ranges		1 0	
.debug_line	164331		
.debug_str	67848		
.comment		1 0	
.ARM.attributes			
.debug_frame			
Total	1525125		
/home/specter/sam	mba/WIFI_RTL8720D_	_sdk6.0a/project	/realtek_amebaD_cm0_gcc_verification/asdk//toolchain/linux/asdk-6.4.1/linux/newlib/bin/arm-none-eabi-size -tradix=10
/home/specter/sam	nba/WIFI_RTL8720D_	_sdk6.0a/project	/realtek_amebaD_cm0_gcc_verification/asdk/image/target_img2.axf
text data	bss dec	hex filename	
92836 30988		le3bc /home/spe	cter/samba/WIFI_RTL0720D_sdk6.0a/project/realtek_amebaD_cm0_gcc_verification/asdk/image/target_img2.axf
92836 30988		le3bc (TOTALS)	
====== Image	Into DEC ======		
rm -f -f ./build/	/ram/*.o		
======= linkei	r 1mg2 end ======	====	
====== Image	manipulating star	rt =======	
/nome/specter/sam	nba/wIFI_RFL8/20D	_sake.Ua/project	/realter amedau cmu_gcc_verification/asdk/gnu_utility/prepend_header.sn /nome/specter/samba/wiri_kiu8/200_sdk6.0a/project/
realter_amebab_cm	nu_gcc_verificatio	on/asak/image/ra	m_2.binram_image2_text_start /nome/specter/samba/wiF1_KTL8/20D_sdk6.Ua/project/reaitek_amebaD_cmu_gcc_verification/
asok/image/target	c_img2.map		
/none/specter/san	IDA/WIFI_RIL6/20D	sake.ua/project	realter amedal cino geo verification/asur/gnu utility/prepend neaderisi /nome/specter/samba/wiri kito/zob sake.od/project/
realter amedal ch	nu gec_verilleatic	on/asok/image/xi	p_image2.biniiash_text_start /nome/specter/samba/wiFi_Kit6/20D_sdx6.0a/project/reaitex_amebab_cm0_gdc_veriiication/
asuk/image/carget	L_ING2.Map	adleC On Immediatel	(maltel ametal and any unification (and) (mu utility (anonand bander at (bare (anostar (ameta) WTRT DRT07200 ad)()) (anoing)
realtok amehaD g	mDa/WIFI_RIL6720D	_suke.0a/project	/ realter allebab church get verification/asuk/ght utility/prepend neader.sh / home/specter/samba/wirf_hib/zob_sub/ght/set/specter/samba/wirf_hib/zob_sub/ght/specter
aption (and) (image	(tanget img2 man	JII/ asuk/ Illiage/ Ia	Tecencion.bin
cation/asuk/intage	s/carget_img2.map	100D address On Jones	icat (westbolk ameter and and unvitigation (and) (image (uin image) approach bin (bowe (appeter (amba /WTPT DUT 0700) add() or (apple
cat /nonle/speccer	D gm0 ggg worifig	120D_Suke.0a/pro	ject/rearrew_amebab_cmo_gcc_verification/asuk/image/xip_image_prepend.in/ /nome/specter/samba/wiri_kib/z0/_suk6.u/proje
image2 all him	D_cmo_gec_verifica	acton/abuk/inage	/fam_z_brepend.bin > /nome/specter/samba/wiri_Kino/zob_suko.va/project/reartex_amebab_cmo_gcc_verification/asuk/image/kmo_
/home/spoctor/sam	who WIFT PTI 9720D	edte 0a (project	(realter amobal cml acc verification/acd//mu utilitu/aad ch /bame/enector/eamba/WIEI PT19720D edv6 0a/necioct/realter amo
hall cm0 ccc verit	fication/asdk/imag	no/km0 image2 al	learces_amebab_cmc_gcc_verificacion/abdx/gnu_ucritcy/pad.on /nome/speccer/samba/wrri_krb0/20D_sdk0.0a/projecc/rearces_ame
Tmage	manipulating end	gey kino_inagez_ai.	
make[1]: Leaving	directory '/home	/specter/samba/W	TFT RTL8720D sdk6.0a/project/realtek amebaD cm0 gcc verification/asdk'
Douring	/ Holder	, outdour, n	

表单编号: B&T-QR-EN-002 版本: B2 保存期限: 5年 生效日期: 2014 年 7 月 4 日 第 3 页 共 12 页



生成文件在 sdk\project\realtek_amebaD_cm0_gcc_verification\asdk\image 目录下

:hared (\\192.168.229.128) (Y:) > WIFL_RTL8720D_sdk6.0a > project > realtek_amebaD_cm0_gcc_verification > asdk > image

名称 ^	修改日期	类型	大小
🗋 km0_boot_all.bin	2019/6/4 15:02	BIN 文件	5 KB
km0_image2_all.bin	2019/6/4 15:02	BIN 文件	96 KB
km0_km4_image2.bin	2019/6/4 15:10	BIN 文件	544 KB
	2010/01/145.02	r Post and Anna	4 120

2.4.2 编译 km4

进入 sdk/project/realtek_amebaD_cm4_gcc_verification 执行 make all 编译成功结果如下

.ram image2.text	102488	268455968	
.ram image2.data		268558456	
.ram image2.bss	44164	268561536	
.ram image2.nocache.data		268605700	
.ram heap.data	262144	268629632	
.xip image2.text	352432	234881056	
.debug info	6210683		
.debug abbrev	214011		
.debug loc			
.debug aranges	23848		
.debug ranges	57960		
.debug line			
.debug str			
.comment			
.ARM.attributes			
.debug frame			
.stabstr			
Total			
/home/specter/samba/WIFI	RTL8720D s	dk6.0a/project/	realtek amebaD cm4 gcc verification/asdk//toolchain/linux/asdk-6.4.1/linux/newlib/bin/arm-none-eabi-size -tradix=10
/home/specter/samba/WIFI	RTL8720D s	dk6.0a/project/	realtek amebaD cm4 gcc verification/asdk/image/target img2.axf
text data bss		hex filename	
454920 289188 44164		0730 /home/spec	ter/samba/WIFI RTL0720D sdk6.0a/project/realtek amebaD cm4 gcc verification/asdk/image/target img2.axf
454920 289188 44164			
======= Image Info DEC			
rm -f -f ./build/ram/*.o			
=========== linker img2 ns			
======== Image manipula	ating start		
/home/specter/samba/WIFI	RTL8720D s	dk6.0a/project/	realtek amebaD cm4 gcc verification/asdk/gnu utility/prepend header.sh /home/specter/samba/WIFI RTL8720D sdk6.0a/project/
realtek amebaD cm4 gcc ve	rification	/asdk/image/ram	2.bin ram image2 text start /home/specter/samba/WIFI RTL8720D sdk6.0a/project/realtek amebaD cm4 gcc verification/
asdk/image/target img2.ma			
/home/specter/samba/WIFI	RTL8720D s	dk6.0a/project/	realtek amebaD cm4 gcc verification/asdk/gnu utility/prepend header.sh /home/specter/samba/WIFI RTL8720D sdk6.0a/project/
realtek amebaD cm4 gcc ve	erification	/asdk/image/xip	image2.bin flash text start /home/specter/samba/WIFI RTL8720D sdk6.0a/project/realtek amebaD cm4 gcc verification/
asdk/image/target img2.ma			
cat /home/specter/samba/W	VIFI RTL872	OD sdk6.0a/proj	ect/realtek amebaD cm4 gcc verification/asdk/image/xip image2 prepend.bin /home/specter/samba/WIFI RTL8720D sdk6.0a/proje
ct/realtek_amebaD_cm4_gcc	verificat	ion/asdk/image/	ram_2_prepend.bin > /home/specter/samba/WIFI_RTL8720D_sdk6.0a/project/realtek_amebaD_cm4_gcc_verification/asdk/image/km4_
image2_all.bin			
/home/specter/samba/WIFI_	RTL8720D s	dk6.0a/project/	realtek_amebaD_cm4_gcc_verification/asdk/gnu_utility/pad.sh /home/specter/samba/WIFI_RTL8720D_sdk6.0a/project/realtek_ame
baD_cm4_gcc_verification/	/asdk/image	/km4_image2_all	
/bin/sh: 1: [: NONE: unex	spected ope		
======== Tmage manipula	ating end =		

生成文件在 sdk\project\realtek_amebaD_cm4_gcc_verification\asdk\image 目录下

ared (\\192.168.229.128) (Y:) > WIFI_RTL8720D_sdk6.0a > project > realtek_amebaD_cm4_gcc_verification > asdk > image

名称	修改日期	类型	大小
APP.trace	2019/6/4 15:10	TRACE 文件	0 KB
cmse_implib.asm	2019/6/4 15:09	ASM 文件	1 KB
🖬 cmse_implib.lib	2019/6/4 15:09	对象文件库	45 KB
🐔 cmse_implib.txt	2019/6/4 15:09	EditPlus 文本 (.txt)	1 KB
] km0_km4_image2.bin	2019/6/4 15:10	BIN 文件	544 KB
km4_boot_all.bin	2019/6/4 15:09	BIN 文件	4 KB
km4_image2_all.bin	2019/6/4 15:10	BIN 文件	448 KB
km4_image3_all.bin	2019/6/4 15:09	BIN 文件	1 KB
🐔 obj list.txt	2019/6/4 15:10	EditPlus 文本 (.txt)	12 KB

3. 串口下载

将 KMO 和 KM4 都编译完成后可以使用串口将编译好的固件下载到模块。

串口下载软件使用 sdk\tools\AmebaZ\Image_Tool\ImageTool.exe 工具

硬件需要用 USB 转 TLL 串口连接模块的 log 串口(LOG_TX(PA7),LOG_RX(PA8))进行下载,接口如下,左侧为模块,右侧为开发底板(开发底板上有两个丝印的看斜杠右侧的丝印)





串口下载需要模块进入下载模式,进入下载模式的方法如下 如果使用开发底板

- (1) 连接好 VCC/GND 和 LOG_TX/LOG_RX
- (2) 按住右边的按键不要松开
- (3) 按下左边的复位按键
- (4) 松开右边的 LOG_TX, 此时模块进入烧录模式



(5) 此时为了检查模块是否处于下载模式可以打开串口工具,波特率 115200,8,N,1,此时用 16 进制显示,如果看到串口如下图一样,不断接收到数据,则表示进入的烧录模式,如果模块打印正常的启动 log,这表示没有进入烧录模式,需要重复上述操作,直到进入烧录模式。





如果使用模块

(1) 首先将 LOG_TX 用 2k 电阻下拉到地 (如果不用电阻下拉部分串口会卡死导致无法同分,部分串口无影响,串口卡死后需要松开 LOG_TX 然后重新插拔串口)

- (2) 保持 LOG_TX 下拉,给模块上电或者按下复位键(丝印为 EN,低电平触发)
- (3) 松开 LOG_TX
- (4) 检查模块是否进入烧录模式和上面使用开发底板的方法相同

当模块进入下载模式之后打开烧录软件 sdk\tools\AmebaZ\Image_Tool\ImageTool.exe

- (1) 点击 Chip Select->AmebaD(8721D) 选择芯片
- (2) 选择要烧录的文件

烧录文件有3个

KM0boot:

sdk\project\realtek_amebaD_cm0_gcc_verification\asdk\image\km0_boot_all.bin
KM4boot:

sdk\project\realtek_amebaD_cm4_gcc_verification\asdk\image\km4_boot_all.bin
KM4image:

sdk\project\realtek_amebaD_cm4_gcc_verification\asdk\image\km0_km4_image2.bin (3) 选择串口然后点击 OPEN 打开串口, 然后点击 download 下载



	aZ((87108) Security (8721D)	_	
CO	OM :	COM2 ~ Baudrate: 1500000	~	Open
Flas	h Er	.a26		
Ad	ldr:	0x08000000 Size: 4	KB	Erase
_				
Flas	h Do	wnload		01d Layout
Γ		Image Path		Address ^
	\checkmark	Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08000000 KM0boot
		system.bin	Browse	0x08003000
	\checkmark	Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08004000 KM14boot
		Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08006000KM4image
		Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08106000
			Browse	
			Browse	
			Browse	
-			Browse	
			Browse	~
二载	ڈ اد	og 如下		
、载	ڈ اد	og如下		
、载	ڈ اد	og 如下		
、载	ڈ Ic	og 如下		
、载	دَ اد	og 如下		
、载	ڈ اد	og如下		
、载	ڈ اد	og 如下		
、载	ڈ اد	og 如下		
、载	ڈ اد	og 如下		
、载	دً اد	og 如下		



<pre>Juded Generate Encrypt Security Serial COM: COMM3 Baudrate: 1500000 Open Flash Brase Addr: Dx08000000 Size: 4 KB Erase Flash Download</pre>	nip Se	lect				
Serial COM: COM3 Baudrate: 1500000 Open Flash Brase Addr: Ox08000000 Size: 4 KB Erase Addr: Ox08000000 Size: 4 KB Erase Flash Bownload 01d Layout 01d Layout Image Path Address 0x08000000 System.bin Browse 0x08000000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Derowse Derowse Browse Browse Browse Derowse	nload	Gener	ate Encrypt Security			
COM: COM3 Baudrate: 1500000 Open Flash Erase Addr: Ox08000000 Size: 4 KB Erase Addr: Ox08000000 Size: 4 KB Erase Flash Bownload 01d Layout 01d Layout Image Path Address 0x08000000 System.bin Browse 0x08000000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08000000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08000000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08006000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08006000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08006000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08006000 Browse Browse Browse 0x08006000 Download 00:18.514 IMG3 image is being sent Image is being sent Image is being sent IMG4 image is being sent Image is being sent successfully! Image is uscessfully!	Se	rial				
Flash Erase Addr: Dx08000000 Size: 4 KB Erase Flash Download		COM:	COM3 ~ Baudrate: 1500000	~	Open	
Addr: Ox08000000 Size: 4 KB Erase Flash Bownload	Fl	ash Er	ase			
Flash Download Old Layout Image Path Address Y'\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Y\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Y\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Y\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Y\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Drowse Browse Browse Browse Brows		Addr:	0x08000000 Size: 4	КВ	Erase	
Image Path Address Y:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08000000 Y:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08006000 Browse Browse 0x08106000 Browse Browse Browse 0x08106000 Browse Browse Browse 0x08106000 Frowse Browse Browse Frowse Frowse Frowse Frowse Browse Browse Frowse Frowse </td <td>Fl</td> <td>ash Do</td> <td>wnload</td> <td></td> <td>🗌 Old Lay</td> <td>/out</td>	Fl	ash Do	wnload		🗌 Old Lay	/out
Image is being sent Image is being sent Image is being sent Image is set successfully! Image is set successfully!			Image Path		Address	^
Image is being sent Image is being sent Image is being sent Image is set successfully! Image is set successfully!			Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08000000	
W:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08004000 W:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08006000 V:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse 0x08106000 Browse Browse 0x08106000 DWG4: Browse 0x018.514 DWG3 image is being sent IMG4 image is being sent IMG4 image is being sent successfully! MG4 image is seen sent successfully! All images are sent successfully! Image			system.bin	Browse	0x08003000	
W:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Ox08006000 W:\gitHome\Wireless\WIFI\WIFI_RTL8720 Browse Browse Browse Browse Browse Download O0:18.514 MG4 image is being sent DMG3 image is being sent MG4 image is being sent MG4 image is being sent successfully! MG4 image is being sent successfully! All images are sent successfully! MG4 image is being sent successfully!			Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08004000	
Wieless/WIFI/WIFI_RTL8720 Browse Browse Browse Browse Browse Browse Browse Download 00:18.514			Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08006000	
Image is being sent Image is seen sent successfully! Image is gent successfully! Image is gent successfully!			Y:\gitHome\Wireless\WIFI\WIFI_RTL8720	Browse	0x08106000	
Image is being sent Image is seen sent successfully! Image is gent successfully! Image is gent successfully!				Browse		
Browse Browse Browse Browse Browse Browse Download 00:18.514				Browse		
Browse Download Download 00:18.514				Browse		
IMG4: Browse IMG4: Download 00:18.514 IMG3 image is being sent IMG4 image has been sent successfully!				Browse		
DWG4: Download 00:18.514 DG3 image is being sent DMG3 image has been sent successfully! DMG4 image has been sent successfully! All image has been sent successfully!			-	Browse		~
Download 00:18.514 IMG3 image is being sent IMG4 image is being sent IMG4 image is being sent IMG4 image has been sent successfully! IMG4 image set successfully! Image has been sent successfully!		IMG4:				
IMG3 image is being sent IMG3 image has been sent successfully! IMG4 image is being sent IMG4 image has been sent successfully! IMG4 image ser sent successfully! IMG4 image are sent successfully!			Download		00:18.	514
IMG3 image is being sent IMG3 image has been sent successfully! IMG4 image is being sent IMG4 image has been sent successfully! All image har successfully!						
DMG3 image is being sent DMG3 image has been sent successfully! DMG4 image is being sent DMG4 image has been sent successfully! All images are sent successfully!	-					
IMG4 image is being sent IMG4 image has been sent successfully! All images are sent successfully!	IMG	3 imag 3 imag	e is being sent e has been sent successfully!			^
IMG4 image has been sent successfully! All images are sent successfully!	IMG	4 imag	e is being sent			
All images are sent successfully:	IMG	4 imag	e has been sent successfully!			
COM3 is closed	CON	3 is c	s are sent successfully: losed			

如果 log 卡死在 Uart download server has started...

这个一般是因为没有进入串口下载模式的原因,首先确认串口是否进入了串口下载模式。

4. 测试

下载完成后接上 log 串口(LOG_TX,LOG_RX,波特率 115200)可以正常打印 log,测试 ATW?指令测试指令是否可以正常执行,注意 AT 指令以回车换行结束。



通用項目 車口设置 显示 发送 多字符串 小工具 報助 联系作者 ▲PCB打样降至每款5元版丰包邮可选杂色: [萬立创窗网] [ATW7]: AT_WLAN_INFO_ WIFI wlanO Status: Running [r]Lk vlan_statistic] tr stat: rx_packets=0, tx_dropped=0, tx_bytes=0 [r]Lk vlan_statistic] rs stat: rx_packets=0, tx_dropped=0, rx_overflow=0 [r]Lk vlan_statistic] ass_skbdat_used_num=1, skbut_used_num=0 [r]Lk vlan_statistic] max_skbut_used_num=1, skbut_used_num=0 [r]Lk vlan_statistic] max_skbut_used_num=10 WIFI vlanO Setting: MODE => OIS e0 :e0 :e0 :e0 :e0 :e0 :e0 :e0 :e0 :e0	L SSCOM V5.13.1 串	口/网络数据;	周试器,作者:大	e下丁.2618	058@gg.co	m. QQ群: 52502449(最新	新版本)	9 <u>44</u>		×
ATW?]: _AT_WLAN_INFO_ WIFI wlan0 Status: Running [rldt wlan, statistic] tr. stat: tr.packets=0, tr. dropped=0, tr. bytes=0, rr. overflow=0 rldt wlan, statistic] rr. stat: tr.packets=0, urr.dropped=0, tr. bytes=0 rldt.wlan, statistic] rr. stat: tr.packets=0, urr.dropped=0, tr. bytes=0 rldt.wlan, statistic] rr.stat: tr.packets=0, urr.dropped=0, tr. bytes=0 rldt.wlan, statistic] max_thing: MODE => STATION SECURITY => OPEN PASSNORD => Interface (wlan0) MAC => 00::0:0:0:0:0 IF => 192.168.1.80 GW => 192.168.1.80 GW => 192.168.1.80 GW => 115200 MAC => 00::0:0:4:67:00:00 IF => 192.168.1.1 MEM => 00::0:0:4:67:00:00 IF => 192.168.1.1 MEM => 00::0:0:0:1 IF => 192.168.1.1 MEM => 00::0:0:0:1 IF => 192.168.1.1 MEM => 00::0:0:0:0:1 IF => 192.168.1.1 MEM => 0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:	通讯端口串口设置	記示发送	多字符串 /	小工具 帮助	联系作者	▲PCB打样降至每款5元	顺丰包邮可选杂色!	【嘉立创官网	1	
<pre>WIFI wlan0 Status: Running [rltk wlan0 Status: Running [rltk wlan1 statistic] tx stat: tx_packets=0, tx_dropped=0, tx_bytes=0 rrtk wlan_statistic] rx stat: tx_packets=0, tx_dropped=0, rx_bytes=0 rrtk wlan_statistic] rx_statistic_packets=0, tx_dropped=0, rx_bytes=0 rrtk=vlan_statistic] rx_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0 rrtk=vlan_statistic_packets=0, rx_bytes=0, rx_byt</pre>	[ATW?]: _AT_WLAN_INFO_									^
[r]tk, wlan_statistic] tx stat: tx_packets=0, tx_dropped=0, tx_bytes=0 [r]tk, wlan_statistic] tx stat: tx_packets=0, tx_dropped=0, tx_bytes=0, rx_overflow=0 [r]tk, wlan_statistic] tx stat: tx_packets=0, current heap free size=170728 [r]tk, wlan_statistic] max_skbbuf_used_num=1, skbbuf_used_num=0 wITI wlan Setting: MDE => STATION SECURITY => OPEN PASSWORD => Interface (wlan0) MC => 00:e0:4e:87:00:00 MC => 00:e0:4e:87:00:00 MC => 192:168.1.40 GW => 192:168.1.1 DMEM] After do omd, available heap 170728 * * motel Silicon Labs CF210x U = HxUB_TAR_MANDELER, abuty and abuty	WIFI wlanO Status: Run	ning								
WIFT wlan0 Setting: MODE → STATION SSID → CHANNEL → 1 SECURITY → OPEN PASSWORD → Interface (wlan0) MAC → 00:e0:40:87:00:00 IF → 192:168.1.80 GW → 192:168.1.1 IMENX After do omd, available heap 170728 # After do omd, available heap 170728 # MIDE ⊂ COM3 Silicon Labs CP210x U: ► HEX显示 (#存動据) @ XiBLD C FITS F DTR 波特率 115200 - INTR F DTR 波特率 115200 - NTY? None ATT-Thread中国人的开源免费操作系统 ★LR-WiFi,无与伦比8005远距离非标WiFi ★新一代WiF Www.daxia.com S:6 R:857 COM3 E打开 115200bps,81,None,None	[rltk_wlan_statistic] [rltk_wlan_statistic] [rltk_wlan_statistic] [rltk_wlan_statistic] [rltk_wlan_statistic] [rltk_wlan_statistic]	tx stat: tx rx stat: rx min_free_he: max_skbbuf_ max_skbdata max_timer_u:	_packets=0, _packets=0, ap_size=1603 used_num=1, _used_num=1, sed_num=10	tx_dropped=0 rx_dropped=0 20, current skbbuf_used_ skbdata_use	, tx_bytes= , rx_bytes= heap free s num=0 :d_num=0	0 0, rx_overflow=0 ize=170728				
MODE => STATION SSID => CHANREL => 1 SECURITY => 0PEM PASSWORD => Interface (wlan0) MAC => 00:e0:4c:87:00:00 IF => 192:168.1.80 GW => 192:168.1.1 [MEM] After do omd, available heap 170728 # 请除窗口 打开文件 第10号 COM3 Silicon Labs CP210x U: minini戰和分包显示, 超时时间 20 ms 第1 字节 至 末尾 マ 加校验 None ● Ximen PATS F DIR 波特率 115200 IT***********************************	WIFI wlan0 Setting:									
Interface (vlan0) IF => 192.168.1.80 GW => 192.168.1.1 IMECTION After do omd, available heap 170728 # 請除窗口 打开文件 「第一日」 第日日」 「第日日」 第日日」 「日田文件」 HEX显示 保存数据 「按山串口」 「日田文件」 HEX显示 保存数据 「日田文学 「日田本学 「日田本学 「日田本学	MODE => STATION SSID => CHANNEL => 1 SECURITY => OPEN PASSWORD =>									
NAC => 00:e0:4c:87:00:00 IP => 192:168.1.80 GW => 192:168.1.80 GW => 192:168.1.1 DMEME After do omd, available heap 170728 # 方除窗口 打开文件 发送文件 停止 青发送区 □ 最前 □ English 保存参数 扩展 - 端口号 COM3 Silicon Labs CP210x U → HEX显示 保存数据 □ 接收数据到文件 □ HEX发送 □ 定时发送: 1 ms/次 ☑ Junep抽拾行。 ● 美词串口 ◆ 更多串口设置 □ Junep抽拾行。 ● 美词串口 ◆ 更多串口设置 □ Junep抽拾行。 ● 第5 ☑ DTR 波特季 115200 ▼ IW? ATT F ☑ DTR 波特季 115200 ▼ IW? AfterMakErscomschafter ★ KIT-Thread中国人的开源免费操作系统 ★LR-WiFi,无与伦比SIMG远距离非标WiFi ★新一代WiF www.daxia.com S:6 R:857 COM3 E打开 115200bps,8,1,None,None CTS	Interface (wlan0)	18								
DNEM] After do omd, available heap 170728 # 請除窗口 打开文件 第次窗口 打开文件 第次窗口 打开文件 第次窗口 打开文件 第次窗口 打开文件 第次窗口 打开文件 第日号 COM3 Silicon Labs CP210x U. 第日日 Comma 指示 第日日 Comma 指示 第日日 Comma Labs CP210x U. ● 美胡串口 Comma 指示 ● 美胡串口 Comma 指示 ● 大胡串口 Comma 指示 ● 日本 大田本 ● TH 波特室 115200 「 ● TW? ● THT ● 大田本 ● THT ● THT <td< td=""><td>MAC => 00:e0:4 IP => 192.168 GW => 192.168</td><td>c:87:00:00 .1.80 .1.1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	MAC => 00:e0:4 IP => 192.168 GW => 192.168	c:87:00:00 .1.80 .1.1								
諸除窗口 打开文件 发送文件 停止 清发送区 □ 端口号 C0M3 Silicon Labs CP210x U: HEX显示 保存数据 接收数据到文件 HEX发送 定时发送: ms:/次 ✓ 第口号 C0M3 Silicon Labs CP210x U: HEX显示 保存数据 接收数据到文件 HEX发送 定时发送: 1 ms:/次 ✓ ✓ 第15 ✓ 115200 I I I I I ✓ ✓ A7 更好地发展Sscom公共件 115200 I I I I ✓ ✓ A7 更好地发展Sscom公共件 I I I ✓ ✓ A7 更好地发展Sscom公共作 ✓ ✓ ✓ A7 更好地发展Sscom公共作 ✓ ✓ ✓ A7 ✓ ✓ ✓ ✓ A7 ✓ ✓ ✓ ✓	[MEM] After do cmd, av	ailable heap	p 170728							
請除窗口 打开文件 打开文件 送 送文件 停止 清发送区 「最前 「 English 保存参数 扩展 — 端口号 [C0003 Silicon Labs CP210x U: ▼ HEX显示 保存数据 [接收数据到文件] [接收数据到文件] [HEX发送 「 定时发送: 1 ms/次 ▼ June车换行] mbri间戳和分包显示,超时时间: 20 ms 第1 字节 至 末尾 ▼ Jun校验 None ▼ [115200 ▼] TW? A7 更好地发展SSCOM软件 发送 [115200 ▼] TW? A7 更好地发展SSCOM软件 发送 [115200 ▼] TW? A7 更好地发展SSCOM软件 发送 [115200 ▼] TW? A7 更好地发展SSCOM软件 [115200 ▼] [1487] SSCOM软件 [1520 ▼] [1487] [15200 ▼] [[1489] SSCOM5.13.1] ★嘉立创PCB打样SMT贴片服务: ★RT-Thread中国人的开源免费操作系统 ★LR-wiFi,无与伦比约M远距离非标wiFi ★新一代WiF mww.daxia.com [S.6 R:857 [COM3 已打开 115200bps,8,1,None,None [CTS] [[CTS] [[COM3] [[[
請除窗口 <u>打开文件</u> <u>发送文件</u> 停止 清发送区 「最前 「English 保存参数 <u>扩展</u> <u>端口号</u> COM3 Silicon Labs CP210x U. HEX显示 <u>保存数据</u> <u>接收数据到文件</u> <u>市区次支送</u> <u>市工次支送</u> <u>市助间離和分包显示。超时时间</u> 20 ms 第1 字节 至 末尾 ▼加校验None ● 、初串口 	#									~
端口号 COM3 Silicon Labs CP210x U. → HEX显示 保存数据 → 接收数据到文件 → HEX发送 → 定时发送: 1 ms/次 → 加回车换行 → ● 关闭串口 ◇ 更多串口设置 → 加时间戳和分包显示。超时时间 20 ms 第1 字节 至 末尾 → 加校验 None → ■ RTS → DTR 波特率 115200 → TW? → ATT → DTR 波特率 115200 → TW? → ATT → DTR 波特率 115200 → TW? → ATT	清除窗口 打开文件				发词	送文件 停止 清发送区	□ 最前 □ Englis	h 保存参数	扩展 —	
● 美田串口 王多串口设置 加时间戳和分包显示,超时时间:20 ms第1 字节 至 末尾 ▼加校验 None RTS ▼ DTB 波特率 115200 ▼ TW? TW? 为了更好地发展SSCOM软件 麦 送 【升级到SSCOM5.13.1】★嘉立创FCB打样SMT贴片服务. ★RT-Thread中国人的开源免费操作系统、★LR-wiFi,无与伦比800远距离非标wiFi ★新一代wiFi www.daxia.com S:6 R:857 COM3 已打开 115200bps,8,1,None,None CTS	端口号 COM3 Silicon La	abs CP210x U	E 🗆 🗆 HEX	品 保存数	据 [接收]	数据到文件 [HEX发送 [定时发送: 1	ms/次 加回	回车换行。	
□ RTS ▼ DTR 波特率 115200 ↓ ↓ ↓ 为了更好地发展SSCOM软件 皮 送 ↓ ↓ ↓ 清您注册嘉立创F结尾客户 皮 送 ↓ ↓ ↓ ↓ 【升级到SSCOM5.13.1】 ★素立创FCB打样SMT贴片服务. ★ RT-Thread中国人的开源免费操作系统 ★ LR-wiFi, 无与伦比800远距离非标wiFi ★新一代WiF www.daxia.com S:6 R:857 COM3 已打开 115200bps,8,1,None,None CTS	🛞 关闭串口 👌	更多串口设	置 加时	间戳和分包显	示,超时时间	20 ms 第1 字节 至	末尾 ▼ 加校验 Non	e 💌		
为了更好地发展SSCOM软件 发送 青您注册嘉立创作结尾答户 发影 【升级到SSCOM5.13.1】★嘉立创FCB打样SMT贴片服务.★RT-Thread中国人的开源免费操作系统 ★LR-WiFi,无与伦比8KM远距离非标WiFi ★新一代WiF www.daxia.com S:6 R:857 COM3 已打开 115200bps,8,1,None,None CTS	□ RTS I DTR 波特率	115200	.▼ Intw?						^	
【升级到SSCOM5.13.1】★嘉立创FCB打样SMT贴片服务. ★RT-Thread中国人的开源免费操作系统 ★LR-WiFi,无与伦比8MG远距离非标WiFi ★新一代WiF www.daxia.com S:6 R:857 [COM3 已打开 115200bps,8,1,None,None] [CTS]	为了更好地发展SSCOM软件 请您注册嘉立创U结尾客户	发i	ž						~	
www.daxia.com S:6 R:857 COM3 已打开 115200bps,8,1,None,None CTS	【升级到SSCOM5.13.1】;	★嘉立创PCB打	T样SMT贴片服	务. ★RT-Th	read中国人的]开源免费操作系统 ★LR⊣	YiFi,无与伦比8KM远	距离非标WiFi	★新一代	WiF
	www.daxia.com S:6	R:8	57 C	омз 已打开	115200bps	,8,1,None,None			CT	S

5. J-link 下载

- (1) 安装 jlink 驱动(官方驱动)
- (2) 指定 jlink 路径

修改 sdk/project/realtek_amebaD_cm0_gcc_verification/jlink_script/cm0_jlink.bat 和 Sdk/project/realtek_amebaD_cm4_gcc_verification/jlink_script/cm4_jlink.bat 中的 jlink 路径为 自己安装的路径。

(3) 设置使用 jlink 调试

分别进入 sdk/project/realtek_amebaD_cm0_gcc_verification 和 Sdk/project/realtek_amebaD_cm4_gcc_verification 执行 make setup GDB_SERVER=jlink 设置使用 jlink 为 gdb 调试工具

(4) 硬件连接将 JLINK 的 swd 接线连接好

(5) 执行 dk/project/realtek_amebaD_cm0_gcc_verification/jlink_script/cm0_jlink.bat 连接成功 可以看到如下结果



Target endian: little Connecting to J-Link J-Link is connected. Firmware: J-Link V9 compiled Apr 20 2018 1	^
Hardware: V9.40 S/N: 59402905 Feature(s): RDI, GDB, FlashDL, FlashBP, JF Checking target voltage Target voltage. 3.27 V Listening on TCP/IP port 2331 Connecting to targetConnected to target Waiting for GDB connection	6:47:26 lash

(6) 进入 sdk/project/realtek_amebaD_cm0_gcc_verification 目录执行 make flash 此时开始下 载 km0 代码,下载成功后可以看到如下结果

816 = 0x8b800 817 = 0x10b
estoring binary file ./image/km0_km4_image2.bin into memory (0x822ac to 0x82aac)
reakpoint 1, RtlFlashProgram () at /cygdrive/f/git_svn/Wireless/WIFI/WIFI_RTL87200_sdk6.0a/project/realtek_ameba0_cm0_gcc_verification/asdk/flashloader/rtl_flash_download.c:88 ILESTATAD08518 = 0x86000 819 = 0x86000 820 = ox10c estoring binary file ./image/km0_km4_image2.bin into memory (0x822ac to 0x82aac)
reakpoint 1, RtlFlashProgram () at /cygdrive/f/git_svn/Wireless/WIFI/WIFI_RTL87200_sdk6.0a/project/realtek_ameba0_cm0_gcc_verification/asdk/flashloader/rtl_flash_download.c:88 sasd["nop"): ILSTARTADDR\$21 = 0x86800 822 = 0x86800 823 = 0x100 mr file / imaga/mg kmt imaga big into ammony (0x872r to 0x872r)
restoring of the synthesis integer with the memory (oxoccae to oxoccae)
reakpoint 1, Rt)FlabProgram () at /cygdrive/f/git_svn/Wireless/WIFI/WIFI_RTL8720D_sdk6.0a/project/realtek_amebaD_cm0_gcc_verification/asdk/flashloader/rt]_flash_download.c:88 ILESTARTADDR524 = 0x87000 825 = 0x4000 826 = 0x10e estoring binary file _/image/km0 km4 image2.bin into memory (0x822ac to 0x82aac)
reakpoint 1, KtFisBhrogram () at /cygdrive/t/git_svn/Wireless/WIFI/WIFI_RIL8720D_sdk6.0a/project/realtek_amebaD_cm0_gcc_verification/asdk/flashloader/rti_flash_download.c:88 sam("nop"); ILESTARTADDR\$27 = 0x87800 828 = 0x8800 829 = 0x10f estoring binary file ./image/km0_km4_image2.bin into memory (0x822ac to 0x82aac)
reakpoint 1, RtlFlashProgram () at /cygdrive/f/git_svn/Wireless/WIFI/WIFI_RTL8720D_sdk6.0a/project/realtek_amebaD_cm0_gcc_verification/asdk/flashloader/rtl_flash_download.c:88 asm("nop"); ump for check
reakpoint 2, RtlFlas <mark>hrvgræs () at /cygdrive///git_Svny</mark> Wireless/WIFI/WIFI_RTL87200_sdk6.0a/project/realtek_amebaD_cm0_gcc_verification/asdk/flashloader/rtl_flash_download.c:120 20 ak[1]: 离开目录''/cygdriver/git_SvnymirtessyntarigRTL87200_sdk6.0a/project/realtek_amebaD_cm0_gcc_verification/asdk ''

(7) 重启模块!!! 注意,这里如果不重新芯片下面的 jlink server 将无法连接,出现闪退
(8) 执行 Sdk/project/realtek_amebaD_cm4_gcc_verification/jlink_script/cm4_jlink.bat(注意:此时 cm0_jlink.bat 也不能关闭,要保持开启)

(9) 进入 Sdk/project/realtek_amebaD_cm4_gcc_verification 目录执行 make flash 此时开始下 载 km4 代码,下载成功后可以看到如下结果





(10) 此时代码就已经全部下载完成了,下载完成后重启芯片就可以正常执行了

6. SDK 目录结构简介

RTL8720D 的内核有两个,一个是 KM0,一个是 KM4, KM0 一般不会修改,我们应用逻辑一般都是添加在 KM4 上的。

以下是一些比较重要的目录(不常用的目录已经删减)

sdk

├ component
├ common
└── at_cmd //AT 指令相关代码
│
— baidu
├── google
│ │ │ ├── wigadget
│ │ │ └── xmodem
└── os //操作系统(freeRtos)
└── project //工程入口
┝— realtek_amebaD_cm0_gcc_verification //KM0 工程(一般不会修改这个目录下的
文件)
└── asdk
│
L— realtek_amebaD_cm4_gcc_verification //KM4 工程
⊢—src
└── main.c //KM4 镜像 main 函数
├ asdk
│ └─ image //编译完成后的 KM4 镜像在这个目录中
└── example_sources //硬件驱动 demo 目录(目录中 mbed 和 raw 是用的两个库
函数实现,用其中一个就好)
表单编号: B&T-QR-EN-002 版本: B2 保存期限: 5 年 生效日期: 2014 年 7 月 4 日
第 11 页 共 12 页

• • •

KM0 的镜像我们一般不会修改,我们的驱动和应用代码一般都是在 KM4 内核上开发的, KM4 内核代码的 main 函数位于 sdk\project\realtek_amebaD_cm4_gcc_verification\src\main.c 中,任务调度系统使用的是 FreeRTOS,在 main 函数最后一行启动的任务调度器,我们可以 根据自己的需要创建任务,添加自己的代码。

硬件驱动主要参考 sdk\project\realtek_amebaD_cm4_gcc_verification\example_sources 中的例子。

应用层主要参考 sdk\component\common\example 和 sdk\component\common\api\at_cmd 中的例子,其中前者是一些 demo 的最简 demo,后者 是现有 AT 指令集的实现。

参考 AT 指令集的实现可以首先查看 AT 指令手册,例如我们要实现连接 wifi 功能,就可 以参考 AT 指令中的 ATPN,然后我们搜索 ATPN 就可以找到如下内容,这个表示 ATPN 对应 的实现函数就是 fATPN。

```
890 #endif
890 #endif
891 ##elif ATCMD_VER == ATVER_2 // uart at command
892 ##if CONFIG_WLAN
893 {"ATPA", fATPA,}, // set AP
894 {"ATPN", fATPN,}, // connect to Network
895 {"ATPH", fATPH,}, // set DHCP mode
896 {"ATPE", fATPE,}, // set static IP for STA
897 {"ATPF", fATPF,}, // set DHCP rule for AP
```

然后我们就可以参 fATPN 的实现或者直接调用这个函数也可以,函数的参数就是我们输入的 AT 指令的参数。

例如 AT 指令执行 ATPN=test01,123456789 , 那 么 就 可 以 直 接 d 调 fATPN("test01,123456789"); 实现相应的 AT 指令功能。

第12页共12页