

GW3323 Printer 开发板 使用说明书

版本: V2.0

© 珠海极海半导体有限公司



目 录

1	编译环境说明	2
1.1	CODEBLOCK(IDE): (CODEBLOCKS-17.12)代码编辑器,编译链接时会调用到 TOOLCHAIN 中提供的工	
	具. 最终生成烧写用的 DCF 文件	2
1.2	把 APP.CBP 拖入 PROJECT 中, 即可打开工程进行编译了	3
1.3	TOOLCHAIN: (RV32-TOOLCHAIN-SETUP_VXXX)包含 RISC-V 编译器, BIN 文件转换工具等	4
1.4	DOWNLOADER 是烧录工具软件兼顾串口打印的功能,可以供开发人员调试,	
	CP210x_WINDOWS_DRIVERS 是 XLINK 烧录器的驱动程序。	4
2	工程的介绍	6
3	下载说明	12
4	例程展示	15
5	空中升级	23
6	常用设置	26
7	版本历史	28



1 编译环境说明

GW3323 此款芯片的开发环境是"Codeblocks.exe",安装包见"codeblocks-20.03mingw-setup.exe". 先安装 CodeBlocks, 再安装 RV32-Toolchain (安装 ToolChain 时, 会向 CodeBlocks 注册配置相关编译环境)

在开发中,我们一般是使用**串口打印**或者 GPIO 口和逻辑分析仪进行调试,暂不支持断点调试和 仿真。

1.1 CodeBlock(IDE): (codeblocks-17.12)代码编辑器, 编译链接时会调用到 ToolChain 中提供的工具. 最终生成烧写用的 dcf 文件.

🌍 Code::Blocks Installat	ion		_		\times
	Choose Install Lo Choose the folder i	cation n which to install Code	Blocks.		
Setup will install CodeBlock and select another folder.	s in the following folde Click Install to start th	r. To install in a differe	ent folder,	click Brow	se
Destination Folder C:\Program Files (x86)	\CodeBlocks		Brow	se	
Space required: 80.9 MB Space available: 58.5 GB					
Nullsoft Install System v3.02,	1	< Back	tall in O	Cance	4,9,9)



Compilers auto-detection		- 🗆 X
Note: After auto-detection, at lea Inspect the list below and change Select you favourite default compi	st one compiler's master path is still the compiler's master path later in ler here:	empty and therefore invalid. the compiler options.
Compiler	Status	 Set as default
Keil CX51 Compiler	Not found	
IAR 8051 Compiler	Not found	
IAR ARM Compiler	Not found	
GDC D Compiler	Not found	
LLVM D Compiler	Not found	
Digital Mars D Compiler	Not found	
GNU Fortran Compiler	Not found	
G95 Fortran Compiler	Not found	
PGI Fortran Compiler	Not found	
No Compiler	Detected	
RISCV32	User-defined	
		~
Current default compiler: RISCV32	2	
	https://blo	g.osdn.net/m0_46(9K)499

1.2 把 app.cbp 拖入 project 中,即可打开工程进行编译了

💮 Code::Blocks Ins	tallation		-	_		×
	Choose Insta	all Location				
	Choose the fi	older in which to instal		KS.		
Setup will install Cod and select another f	eBlocks in the following older. Click Install to st) folder. To install in a tart the installation.	different f	older,	click Brov	wse
Destination Folder						
C:\Program Files	(x86)\CodeBlocks			Brow	se	
Space required: 80.9	9 MB					
Space available: 58.	5 GB					
Nullsoft Install System v	/3.02.1			_		
		< Back	Install	iler ()	Cano	e]499



🐴 main.c [app] - Code:Blocks 20.03	
File Edit View Search Project E	suild Tools Plugins Settings Help
💽 🕒 🖬 🎒 🐁 🤿 💢 🖻 🏨	Q. 9.
i ⇔ ⇒ ≥ ≥ ≥ ≥ ≤	📀 🖬 Debug 🔹 🖪
sglobal>	•
Menagement X	fune motorie fune emdie bap sysie fune emdih appie fune bae mainic
Projects	1 8/2222
😡 Workspace	2 * GR/3323的关机功耗4uA(开PHRKEY或者TO KEY长按也能进入关机状态, func ob.)
🖃 📲 app	3 * 睡眠模式300mA.
⇒ Sources	4 * 用户可使用80xs的sRam,
E-C platform	5
🕮 🎃 bsp	6 #include "include.h"
⇒ inctions	7 volatile u8 Motor_Step=0;
- func.c	
- func_bt.c	3 AF(.user_i.data) //buffer[and/second with texts]-implified texts are 10 up 100 pr DebtsDuf(Duffer[and/second with 100]
func bt dute	10 00 COART_MEDICALICIDENCES, ////////////////////////////////////
- func emd.e	12 AT(.user 3.data)
- func idle.c	13 uint32 t CMD 1D2A buff(40);//=0xa0bytes; #0°å%*# akka *cnd 1b2ac86; -å@°å64
func lowpwr.c	14 AT(.user 3.data)
func motorie	15 uint8 t USART_TX5UF[0x40];//
func usbdev.c	16
₽-@a libs	17 u32 pReadAddr;
= modules	18 u32 pWritwAddr;
É-la bie	15 u32 AnalyLine=0;
⇒ ab link	20 u32 PrintLine ⁻⁰ ;
anne	21 us Linebata[10]; 22 uslatile us notor flage1//2素量合地行
- adv0.c	23 volatile us Heat flage)
blec	24 volatile u8 noto_Dir flag=01//0表示正转。1表示反转
in the lat	25 volatile u8 SendGroupNum-0;
	26
	<pre>27 const uint0 t SoftVersion[0]={'2','0','2','2','0','6','0','2',};</pre>
n a message	28
	29
Connigic Dimain c	30 const wint8_t *bt_rf_gst_param(void)
B D Headers	41 □1 <
0 Dthers	lars & athers
	Codectioners S Search result S Build log P Build messages
	Durput file is Cutput bill app rv22 with rise 10 r 30 KB
	Ostput/bin/bootbuild.bet app
	E:/bt/GE3333_Apex/SIE/projects/Cutput/bin/riscv12-eif-objccor -O binary mon rv52 mon bin goto err
	n: horbesize' voe Au DODE SIZE: See Au
	save file "azo. def" survessful
	E:\bt\GW3323_Apex\SDE\projects\Output\bin)if exist C:\uplead\uplead.bst (call C:\uplead\uplead.bst =D STE93CA2 app.dcf)
	ne ter personal agres para person person person person of () a feature (a) a feature (
6	0 error (a). 0 eerror (a) () ainute (a). 2 eeroed (a))
· · ·	

1.3	ToolChain: (RV32-Toolchain-Setup	_vxxx)包含 RISC-V 编i	泽器, Bin 文	、件转换工具等.
	🔂 RV32-Toolchain-Setup.exe	2021/12/30 20:48	应用程序	24,977 KB

1.4 downloader 是烧录工具软件兼顾串口打印的功能,可以供开发人员调试, CP210x_Windows_Drivers 是 Xlink 烧录器的驱动程序。

🚾 CP210x_Windows_Drivers.rar	2022/3/11 17:08	WinRAR 压缩文件	3,656 KB
www.ioader_v2.7.2.zip	2022/3/11 16:38	WinRAR ZIP 压缩	2,488 KB



Jownloader v2.7.2	ato		×
选项(O) 工具(T) 扫码枪(G) 帮助(H)		Language	置顶
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□			
DownFile D:	-\GW 🕶 📔	• 🖉 •	• •
□ 暫停 🚢 滾动 🗊 全选 🗈 复制 📙 保存 🔹 🔄 格式 ▾ 📑 信息 ▾ 🗔 擦除		1101	、 清空
[COM36]			^
[COM36] 2023/12/14 14:49:03: 扫描中			
[COM36] 2023/12/14 14:49:03:开始:高速模式			
[COM36] 2023/12/14 14:49:03: 程序大小: 306.0 KByte			
[COM36] 2023/12/14 14:49:03: 不校验KEY			
[COM36] 2023/12/14 14:49:03: 蓝牙地址模式: 固定地址			
[COM36] 2023/12/14 14:49:03: 蓝牙地址: 41:42:86:99:00:01			
[COM36] 2023/12/14 14:49:05: 开始下载			
[COM36]			
Hello Platform			
RC2M[0]: 2120984Hz, Counter: 45262			
RC2M[1]: 3256555Hz, Counter: 29479			
startup			
Hello GW3323: 8021003a			
WDT reset			
xcfg init ok			
RC2M[0]: 1975633Hz, Counter: 48592			
bt_uart_isr_en			
ble_get_profile_data			
System configuration completed!			
295			
			~
完成 一〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇	3切 配置		

开发人员一般选中下图中的"开发",可以方便下载后查看打印信息.

Jownloader v2.7.2	— D	\times
选择串口工具(T) 扫码枪(G)选择器(序配置选项 开始下载,里面有全片擦除等可选	Language	置顶
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□		
DownFile D:\	r8: • 📸 • 🤌 🕌	•
💵 暂停 🚢 滾动 📮 全选 🗈 复制 🚽 保存 🔹 🔄 格式 🗸 😁 信息 👻 🗔 擦除	218 🗏	▶ 清空
[SYS] [SYS] 2023/12/13 10:11:29: 打开DCF调试下 <mark>截求进片信息</mark> [SYS] 2023/12/13 10:11:29: 程序大小: 305.0 KByte [SYS]	与出生产用的pr 件或者加密配置 件 ecf	o文 文



2 工程的介绍

将 "GW3323_SDK_V1.0\GW3323_SDK\projects\GW3323.cbp" 文件用 "codeBlocks" 开 发工具打开。

« G	W3323_SDK_V1.0 > GW3323_SDK > p	projects	ٽ <u>~</u>	在 projects 中搜到	素
	へ 名称	修改日期	类型	大小	
	example	2023/2/23 9:59	文件夹		
	Output	2023/2/23 9:59	文件夹		
R	🛞 config.c	2023/1/31 14:25	C 文件	8 KB	
_ * <mark>-</mark>	🛞 config.h	2023/2/13 20:17	H 文件	16 KB	
- *	🝘 GW3323.cbp	2023/2/23 9:31	CBP 文件	12 KB	
_ * ^	GW3323.depend	2023/2/23 10:40	DEPEND 文件	20 KB	
	GW3323.layout	2023/2/23 10:45	LAYOUT 文件	1 KB	
	🝘 GW3323_BT_UART.cbp	2022/12/12 17:00	CBP 文件	20 KB	
诱传	GW3323_BT_UART.depend	2022/12/12 17:01	DEPEND 文件	24 KB	
AL21~	GW3323_BT_UART.layout	2022/12/12 17:37	LAYOUT 文件	8 KB	
	🔫 main.c	2023/2/23 10:25	C 文件	8 KB	
	📑 ram.ld	2022/10/12 17:21	LD 文件	12 KB	
	🔫 user_datas.c	2023/2/14 16:37	C 文件	20 KB	
	🍘 user_datas.h	2023/2/22 15:53	H 文件	8 KB	
间	🛞 xcfg.h	2022/12/6 19:43	H 文件	16 KB	

打开后出现如下界面:

🧼 🕨 🌯 🌚 Debug	\sim	🗄 🕴 🗢 🖻	→ [· · · · · · · · · · · · · · · · · ·	
<global></global>		~		~~~~
anagement	×	main.c		2-6-
Projects		1	finaluda Minaluda bW	
Workspace		2	Finerade include.n	
		2	/*****Printer and Bluetooth transm	ission******/
GW3323		4	finglude "user datas h"	18810114444447
Sources		5	finclude "min out led h"	/*Bluetooth LED#/
🗄 👝 Headers		6	finclude "hsuart transfer h"	/*Serial port*/
🗄 🕞 Others		7	finclude "printer h"	/*printer*/
		8	printing printering	, princi ,
		9	/*************************************	******
		10	#include "charge.h"	,
		11	finclude "dac out.h"	
		12	#include "i2c eeprom.h"	
		13	#include "power sleep wake.h"	
		14	#include "rtc calendar.h"	
		15	#include "saradc kev.h"	
		16	#include "saradc sampling.h"	
		17	<pre>#include "sd card.h"</pre>	
		18	<pre>#include "spil flash.h"</pre>	
		19	<pre>#include "timer capture.h"</pre>	
		20	#include "timer led.h"	
		21	#include "timer pwm.h"	
		22	<pre>#include "uart transfer.h"</pre>	
		23	#include "wdt.h"	
		24	/********	*********/
		25		
		26		
		27	//Start Main function	
		28	int main (void)	
		29		
		30	bsp_sys_init();	/*System initialization*/
		31		
		32	/****Printer and Bluetooth initia	lization******/
		33	77	



左侧三个文件夹分别为放着工程文件的 xxx.c 和 xxx.h 文件、Others 文件夹下是输出下载文件

0	Workspace		
ė	GW3323		
1	🖹 🔁 Sources 🔒 C		
	🗉 🕞 platform		
	🖻 🔁 projects		
	🗄 🔁 example		
	config.c		
_ [main.c		
ī	🗏 🔁 Headers 🔥 h	_	
	🗄 📄 platform		
	🖃 🔁 projects		
	🕀 🔁 example		
	config.h		
	user_datas.h		1
	🗋 xcfg.h		3
	🖃 🔁 Others		٦.
	🖃 🖻 projects下载秸]序	2
	🖃 🕞 Output		
	🕀 🕞 😥 bin		
0	ram.ld		

点击 main.c 可看到所有的示例程序



GW3323_SDK_V1.0 工程文件默认打开蓝牙透传功能和打印机打印固定字符功能,如需使用其 它例程功能,打开对应初始化及示例函数即可。



例:

如需使用 IIC 功能,即可取消对应的 IIC_AT24C01_init()初始化函数和 IIC_AT24C01_exa mple()函数的注释,并注释掉其它不用的初始化函数。



在 bsp_sys.c 里存放系统初始化的一些函数





```
main.c hsuart_transfer.c app.c spp.c bsp_sys.c
  590
               3
  591
  592
   593
          u8 ack_Name[31]={"BT"};
          AT(.text.bsp.sys.init)
  594
          void bsp_sys_init(void)
  595
        ⊟ {
  596
  597
       н
               /*Power-on transmission test serial port*/
  598
               u32 lvdcon = LVDCON;
              printf("Hello GW3323: %x\n", lvdcon);
  599
                                          printf("WKO reset\n");
   600
               if(lvdcon & BIT(18))
                                          printf("VUSB reset\n");
               else if(lvdcon & BIT(17))
   601
   602
               else if(lvdcon & BIT(16)) printf("WDT reset\n");
   603
               else if(lvdcon & 0xf00)
                                          printf("SW reset\n");
  604
  605
               /// config
                                           , /Get
2023
        þ
   606
               if (!xcfg_init(&xcfg_cb, sizeof(xcfg_cb))) {
                                                                      //Get configuration
   607
                  printf("xcfg init error\n");
   608
  609
               else
                  printf("xcfg init ok\n");
   610
   611
   612
               // io init
   613
              bsp_io_init();
  614
   615
               // var init
   616
               bsp_var_init();
   617
   618
               // power init
  619
        Т
               pmu_init((BUCK_CURR_LIMIT_DIS << 7) | BUCK_MODE_EN);</pre>
   620
               adpll_init(0);
   621
               // clock init
   622
               set sys clk(SYS CLK SEL);
   623
             // peripheral init
  624
   625
               rtc_init();
   62.6
               param_init(sys_cb.rtc_first_pwron);
   627
```

关于 SPP 蓝牙的控制在 spp.c 中:







关于 BLE 蓝牙的控制在 app.c 中:





main.c	hsu	art_transfer.c	app.c	spp.c	bsp_sys.c					
160)	11	4				- 2.0	0		
161	L	11		puf[0] = 0x11;	;				
162	2	11		hle_t	x_notify	(gatts_Dat	as_Charact	eristic_b	ase.att_i	ndex, buf, 1);
163	3	11		BtFlo	wControl_	start = 0	;			
164	ł	1111		pri	ntf(" BTH	FlowsContro	ol_Start \	n");		
165	5	11	}							
166	5	// }								
167	7	L-}								
168	3									
169	9	/*Low-po	ower B	luetoo	th*/					
170		extern v	rolati:	le u8	g_ucAT_fl	Lag;				
171	L	extern u	int8_	t pri	nt_pic;					/* Judge the print c
172	2	static u	int8_	t gatt	_callback	t_app (u8 *)	ptr, u16 1	en)		
173	3	⊢ {								
174	÷ .	g_uc	AT_fla	ag = 0	;					
175	5	hsua	irt_dma	a_star	t (HSUT_TF	RANSMIT, (1	uint32_t)p	tr, len);		/*Send the received
176	5	prin	ntf("[\$x] %x	%x%x\r	n",len,ptr	[0],ptr[1]	,ptr[len-	1]); 🚬	/*Test serial port*
177	7									
178	3		.en ==	2)						/*Judge the print
179	9	F {								
180)		if(p	tr[0]	== 0xld &	&& ptr[1] :	== 0x34)			
181	L		p	rint_p	ic = 1;					
182	2	- }								
183	3									
184	ł	retu	irn fa	lse;						
185	5	L- }								
186	5	/******	***/							
187	7									

gw3323 工程文件中放置外设的库函数,自己写函数时需调用这些函数。





3 下载说明

1、用 CP2102 的串口模块,或我司自己的串口模块连接开发板。

极海的 XLink 模块: RX---->接 GW3323 的 PB3;

CP2102 串口模块: TX -- 串 200R -- RX (拉 RX 这根线到主控) --->接 GW3323 的 PB3;



2、串口模块的(GND,RX,3.3V)3根线接到开发板(GND,PB3,3.3V)上,另一端接电脑的USB口。



3、连接串口后,"串口图标变黑",表明串口已接上,否则查看串口硬件问题或者 "CP210x_Windows_Drivers" 驱动安装的问题。

▲ 设备管理器	-	×
文件(F) 操作(A) 查看(V) 帮助(H)		
✓ 🛃 XO-7008		^
> 📷 IDE ATA/ATAPI 控制器		
> 💼 WSD 打印提供程序		
> 🖸 处理器		
> 🔜 磁盘驱动器		
> 🎥 存储控制器		
> 🖻 打印队列		
> 🖻 打印机		
✓ 算 端口 (COM 和 LPT)		
📮 Prolific USB-to-Serial Comm Port (COM40)		
🛱 Silicon Labs CP210x USB to UART Bridge (COM36)		
📮 打印机端口 (LPT1)		
💭 通信端口 (COM1)		
> 📓 固件		



4、 然后再选择工程烧录文件.dcf;再选择开始烧录。如下图:

(1)例程的烧录文件在 GW3323_SDK_V1.0\GW3323_SDK\projects\Output\bin\gw3323.dcf



(2)选择 CP2102 串口在电脑上的串口号

Jownloader v2.7.	2	- 0	×
选项(O) 工具(T)	14码枪(G) 帮助(H)	Language	置顶
📮 串口 🖌 🏺 USB	※配置 ▼ ▶ 开始 ▼ 目 开发		
全部关闭		} - ∌ -	• 🛃
	全选 🗈 复制 🚽 保存 🔻 🔄 格式 🖌 📑 信息 👻 🗔 擦除	0	家 清空
COM36			^
COM40	2		

(3)点击配置 project

Jownloader v2.7.2		_	с x
选项(O) 工具(T) <u>扫码枪(G)</u> 普助(H)		Lang	guage 置顶
🌐 串口 🔹 🔍 USB 🔀 🖓 配置 🔹 开始 👻 🗐 开发	:		99
DownFile D:\桌面\思 管理配置 23		C 🚰 🔸	ۍ 🛃 ד 🌮
	試 ▾ 🚰 信息 ▾ 🗆 擦除		173 式 清空
[COM36] [COM36] 2023 2 Project 日借中 [COM36]			^

(4)点击开发按钮



选项(O) 工具(T) 扫码枪(G) 帮助(H) ■ 串口 • • USB ※ 配置 • ▶ 开始 • ■ 开发	
🟺 串口 🔹 🖗 USB 🔗 配置 🔹 🕨 开始 🔹 🖃 开发	
DownFile D:\{	·打
□ 暂停 描 滚动 🗊 全选 🗈 复制 🛃 保存 🔹 🔄 格式 マ 🖙 信息 マ 🗆 擦	滁

(5)点击开始即可下载程序

Downloader v2.7.2	
选项(O)·工具(T) 扫码枪(G) 帮助(H)、	Language 置顶
DownFile E:\GW3323\GW3323_BT_20221213\app\projects\Output\bin\app.dcf	🗩 - 💌 -
』 暫停 📇 滾动 🗊 全选 🗈 复制 🛃 保存 🔹 🔄 格式 ▾ 😁 信息 ▾ 🗔 擦除	1142 🛒 清空
[2]: [SYS] [COM15] [COM15] 2022/12/21 18:32:44: 用方 [COM15] 2022/12/21 18:32:44: 福序大小: 2022/12/21 18:32:44: 福序大小: 2022/12/21 18:32:44: 福序大小: 2022/12/21 18:32:44: 福牙地址模式: [COM15] 2022/12/21 18:32:44: 蓝牙地址 [COM15] 2022/12/21 18:32:45: 月猫下载 [COM15] 2022/12/21 18:32:45: 月猫下载 [COM15] 2022/12/21 18:32:46: 這方 [CO	E
	·



4 例程展示

1、默认例程(GW3323_SDK_V1.0)是"打印机测试与蓝牙透传",示例程序包括:

打印机输出固定字符:

按压开发版的 SW5 按键, 能控制打印机输出 "Geety" 字符

也可连接蓝牙通过 16 进制指令->0x1d34 控制打印机输出固定字符 "Geety"

蓝牙透传:

蓝牙可与串口通讯,实现手机发送数据可通过高速串口(PB1和PB2)发送到电脑或其它开发板 PB1为开发板的 RX 引脚

PB2 为开发板的 TX 引脚

2、更改例程后要进行编译,"0错误0警告"后才能正常烧录。



- 3、 蓝牙透传主要用到 spp.c(经典蓝牙)和 app.c(低功耗蓝牙)的两个文件, 蓝牙接收到数据将自动 进入中断,我们只需把接收到的数据发送给串口即可实现蓝牙透传。
- 4、spp.c 中,函数支持经典蓝牙协议,可在手机上下载 "spp 蓝牙串口" APP 与开发板进行连接,在程 序中我们将接收到的数据放在 packet 数组中,通过高速串口 PB1 和 PB2 发送到电脑端。
- 5、app.c 中,函数支持低功耗蓝牙协议,可在手机上下载 "BLE 调试助手" APP 与开发板进行连接, 在程序中我们将接收到的数据放在 ptr 数组中,通过高速串口 PB1 和 PB2 发送到电脑端。







6、这样我们只需要在主函数中初始化串口和循环等待串口接收函数即可实现蓝牙的透传功能。波特率



设置为 115200, 也可根据需要进行更改。



mair	n.c hsua	rt_trai	nsfer.c
	32		2.00
	33		clock gate0 cmd(CLOCK GATE0 HSUARTO, ENABLE);
	34		
	35		<pre>gpio_init_structure.gpio_pin = GPIO_PIN_1;</pre>
	36		gpio_init_structure.gpio_dir = GPIO_DIR_INPUT;
	37		<pre>gpio_init_structure.gpio_fen = GPIO_FEN_PER;</pre>
	38		<pre>gpio_init_structure.gpio_mode = GPIO_MODE_DIGITAL;</pre>
	39		<pre>gpio_init_structure.gpio_pupd = GPIO_PUPD_UP;</pre>
	40		<pre>gpio_init (GPIOB_REG, &gpio_init_structure);</pre>
	41		
	42		gpio_init_structure.gpio_pin = GPIO_PIN_2;
	43		gpio_init_structure.gpio_dir = GPIO_DIR_OUTPUT;
	44		gpio_init_structure.gpio_drv = GPIO_DRV_8MA;
	45		<pre>gpio_init (GPIOB_REG, &gpio_init_structure);</pre>
	46		
13	47		gpio_func_mapping(HSUTTXMAP_PB2);
1	48		<pre>gpio_func_mapping(HSUTRXMAP_PB1);</pre>
	49	Г	
	50		hsuart_init_struct.baud = 115200;
	51	11	hsuart_init_struct.baud = 460800; /*Baud rate selection*/
	52	1/-	nsuart_init_struct.baud = 230400;
	53	//	hsuart_init_struct.baud = 256000;
	54		
	55		hsuart_init_struct.tx_mode = HSUT_TX_DMA_MODE;
	56		hsuart_init_struct.rx_mode = HSUT_RX_DMA_MODE;
	57		hsuart_init_struct.tx_stop_bit = HSUT_STOP_BIT_1BIT;
	58		hsuart_init_struct.tx_word_len = HSUT_TX_LENGTH_8b;
	59		hsuart_init_struct.rx_word_len = HSUT_RX_LENGTH_8b;
	~~~		• • • • • • • • • • • • • • • • • • • •





7、打印机输出固定字符程序中我们初始化打印机的各个引脚,并在打印机例程中等待标志位,如按键 按下与蓝牙接收到固定数据,如相关标志位置1,我们让打印机打印"Geehy"字样。

```
/*****Printer and Bluetooth initialization******/
                                         /*LED2 initialization*/
    gpio led init();
                                         /*Serial port initialization*/
    my hsuart init()
                                            500m-
    printer init();
                                         /*printer initialization*/
 /****Other routines Related initialization*****/
11
    gpio_input_init();
11
      Charge_init();
     my_dac_init(DAC_L | DAC_R);
 11
     IIC AT24C01 init();
 11
 11
     power_sleep_wake_init();
 11
      rtc_calendar_init();
 17
     timer3_cap_init(66666);
                                  // Fsrc=1MHz, Fpwm=1KHz
      timer3 init(500000 - 1);
11
11
      timer3_pwm_init(1000 - 1);
 11
      uart2 init(115200);
11
      Adc_key_init();
11
      Adc sampling init();
      sd_disk_init();
11
      spil init(SPIFALSH BAUD);
11
 while (1)
-
        LED_Bluetooth();
                                             /*LED2 is always on after Bluetooth connection*/
                                             /*Wait for reception interruption - receive Bluetoot
        hsuart transfer example();
        printer_example();
                                             /*Print fixed font "Geehy"*/
 153
 154
         void printer_init (void)
 155
 156
             gpio_init_typedef gpio_init_structure;
 157
             spi_init_typedef spi_init_structure;
 158
             uint8_t ii;
 159
 160
            gpio init structure.gpio pin = GPIO PIN 0 | GPIO PIN 1 | GPIO PIN 2 | GPIO PIN 3;
            gpio_init_structure.gpio_dir = GPIO_DIR_OUTPUT;
 161
             gpio_init_structure.gpio_fen = GPIO_FEN_GPIO;
 162
 163
             gpio_init_structure.gpio_mode = GPIO_MODE_DIGITAL;
 164
            gpio_init_structure.gpio_drv = GPIO_DRV_8MA;
 165
 166
            gpio init(GPIOF REG, &gpio init structure);
 167
            gpio_init_structure.gpio_pin = GPIO_PIN_5;
 168
 169
             gpio_init_structure.gpio_dir = GPIO_DIR_INPUT;
 170
             gpio_init_structure.gpio_fen = GPIO_FEN_GPIO;
            gpio_init_structure.gpio_mode = GPIO_MODE_DIGITAL;
 171
 172
            gpio_init_structure.gpio_pupd = GPIO_PUPD_UP;
 173
 174
            gpio_init(GPIOB_REG, &gpio_init_structure);
 175
             gpio_init_structure.gpio_pin = GPIO_PIN_0 | GPIO_PIN_4 | GPIO_PIN_5;
 176
             gpio_init_structure.gpio_dir = GPIO_DIR_OUTPUT;
 177
 178
            gpio init structure.gpio fen = GPIO FEN GPIO;
 179
             gpio_init_structure.gpio_mode = GPIO_MODE_DIGITAL;
            gpio_init_structure.gpio_drv = GPIO_DRV_8MA;
 180
 181
            gpio_init(GPIOE_REG, &gpio_init_structure);
 182
 183
 184
            clock_gatel_cmd(CLOCK_GATE1_SPI1, ENABLE);
 185
                                                                           //CLK
             gpio_init_structure.gpio_pin = GPIO_PIN_3 | GPIO_PIN_4;
 186
 187
             gpio_init_structure.gpio_dir = GPIO_DIR_OUTPUT;
            gpio_init_structure.gpio_fen = GPIO_FEN_PER;
 188
            gpio init structure.gpio mode = GPIO MODE DIGITAL;
 189
 190
            gpio_init_structure.gpio_pupd = GPIO_PUPD_UP;
            gpio_init(GPIOA_REG, &gpio_init_structure);
 191
 192
            gpio_func_mapping(SPI1_MAP_G1);
 192
```



main.c	printer.c
255	$Run_OK = 0;$
256	tmr_cmd(TMR3, ENABLE);
257	while (Run_OK == 0) {
258	<pre>// printf( "Run_OK %d\n", Run_OK);</pre>
259	
260	
261	
262	
263	/*Judge the key and Bluetooth reception flag bit, and print fixed font*/
264	uint32_t picpt_offset;
265	uint8 t print pic = 0;
266	<pre>void printer_example(void)</pre>
267	
268	<pre>if (gpio_read_bit(GPIOB_REG, GPIO_PIN_5) == RESET &amp;&amp; print_pic != 1) {</pre>
269	<pre>print_pic = 1;</pre>
270	
271	22
272	if(print_pic)
273	
274	<pre>step = 0;</pre>
275	<pre>picpt_offset = 0;</pre>
276	while(step < 960)
277	
278	<pre>PrinterLatch_RESET();</pre>
279	nrinthead conductes (nic nt+nicht offset 48).

8、程序中还有蓝牙指示灯函数,通过判断蓝牙是否连接来展示不同的状态,当蓝牙尚未连接时,LED2 指示灯以 500ms 的频率闪烁,当蓝牙连接到开发板后,LED2 常亮提示连接成功。







9、整体连接图





10、 程序下载完成后即可在电脑上打开串口助手(以 SSCOM 软件为例),设置比特率 115200,选择 端口号,勾选 HEX 显示。使用手机软件(以 SPP 为例)连接开发板蓝牙,选择 hex 发送格式,输入数据即可在电脑端看到。

▲ (★有新版本V5.13.1★)SSCOM V5.12.2	
	바 가그릇 하세 地立BFCU314
2 34 12 34 12 34 12 34 12 34 12 34 12 34 12 3	12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34 12 34
清除窗口   打开又件	
嵩口号 COM40 Prolific USB-to-Seri 🚩 🔽	7 HEX显示 保存数据  □ 接收数据到文件  □ HEX发送 □ 定时发送:1000 ms/次 □ 加回车排
◎ 羊油串口 ▶ 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 ▼
	7172727\n
RIS DIR 波特率: 113200 -	
了更好地发展SSCOM软件 发送	
【开级到V5.13.1】★合宙高性价比4G模块值;	傳一讯 ★KT-Thread中国人的升源免费操作系统 ★新一代Wifi芯片兼谷8266支持KT-Thread ★8KM
↓ 41:42:86:99:00:01 ★月:42:86:99:00:01 ★月:42:86:99:00:01	
us-ascii 🗸 🔽 显示发送 🔽 自动滚动	
Ⅱ 合 回···  ■ 收:50 ■ 皮:成功:62 失效:0	^
5:58:06.016> 12 34	
5:58:06.2/3> /2 34	
5:58:06.576> 12 34 5:58:06.576> 12 34	
5:58:06.742>12.34	
5:58:07.142>12.34	
5:58:07.441> 12 34	
5:58:08:075> 12 34	
5:58:08.341> 12 34	
5:58:08:695> 12 34	
5:58:09.108> 12 34	
5:58:09.426> 12 34 5:58:09.426> 12 34	
5:58:09.578> 12 34	
5:58:09.990> 12 34 5:58:09.990> 12 34	
5:58:10.141> 12 34	
5:58:10.290> 12 34 5:58:10.891> 12 34	
5:59:10.458> 417172727	
5:59:12.379> 417172727	
5:59:13:448> 417172727	
5-59-13 768> 417172727	
5:59:14.159> 417172727	
122//	×
1234 反达	× ·
hex v 🕓 🛷	
ex ~ () 《 ] 循环发送 延时(ms); 1000	



11、 我们也可发送 hex 格式的 1d34 数据,即可看到打印机打印出"Geehy"字符,也可通过 SW5 按 键控制打印机输出"Geehy"字符。



15:58:09.990> 12 34			
15:58:10.141> 12 34			
15:58:10.290> 12 34			
15:58:10.891> 12 34			
15:59:10.458> 417172727			
15:59:12.379> 417172727			
15:59:13.448> 417172727			
15:59:13.768> 417172727			
15:59:14.159> 417172727			
16:05:48.316> 1D 34			
1 d <mark>3</mark> 4		发送	≽
hex ~ 🕓 💋	,		
□ 循环发送 延时(ms): 1000			



# 5 空中升级

名称	修改日期	类型	大小
Boards	2023/8/31 14:12	文件夹	
Documents	2023/9/19 20:31	文件夹	
	2023/9/14 11:35	文件夹	
Libraries	2023/9/14 11:35	文件夹	
Package	2023/8/31 14:13	文件夹	
🔛 GW3323_fota_forAPP.rar	2023/6/7 10:50	WinRAR 压缩文	12,496 KB
📄 版本记录.txt	2023/9/19 20:32	文本文档	8 KB
📄 版本记录.txt.bak	2023/9/13 16:20	BAK 文件	8 KB

1、下载如图所示升级 APP 安装包到电脑, 解压后, 进入如下目录 app_src\FOTA_Android_Library\apk



2、将.apk 文件发送到手机,手机下载后安装软件,软件安装好后,如下图所示



3、打开软件后,选择 BLE OTA 或 SPP OTA 即可进行 GW3323 的空中软件升级,以 BLE OTA 为例, 具体步骤如下图所示。





选择自己的蓝牙名称

BLE设备扫描	停止扫描
N 58_BL	LE
未知设备 3B:C3:39:69:71:BC	36
出现版本号,点击选择文件	
BLE OTA (KM-360SE-BA58 _ BL···	选择文件
	Î
版本号: 0.0.9.0	

选择自己制作的.fot 文件,点击 SELECT 按键

AB OTA Demo	根目录
(storage/emulated/0/DingTalk テ ESC引印用1例.txt	
<b>2023/023(/).fot</b>	26
<b>4</b> (0702) E020 2023/128-2 fot	
9 - ota.bin	112
SELECT	



#### 点击开始升级,等待进度条结束后,重启设备即可



开始升级



在下载工程界面点击如下图所示的图标

Jownloader v2.7.2	-		×
选项(O) 工具(T) 扫码枪(G) 帮助(H)	L	.anguage	置顶
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	-		_
	• 🖻	• 🖉	•
🖩 暂停 🚢 滾动 🗊 全选 🗈 复制 🔛 保存 🔻 🔄 格式 👻 😁 信息 👻 🗔 擦除		782	<b>家</b> 清空
MAC:41 42 E8 46 C0 23			^

选择文件格式,设置文件名后,保存即可





# 6 常用设置

1、查找所有文件(ctrl+F)

Find Find in files		
Text to search for:		Options
HSUT_TRANSMIT		<ul> <li>✓ Limit to:</li> <li> <ul> <li> <u>W</u>hole word                 <u>Start word</u> <u>Start file</u> <u>Match case</u> <u>Regular expression</u> <u>Multi-line search</u> <u>Dolete provinue coorch regular</u> </li></ul> </li> </ul>
Scope <u>O</u> pen files <u>P</u> roject files <u>W</u> orkspace files <u>S</u> earch path	Project to search in Project: app Target: All project fil	es

5、恢复窗口视图





#### 6、注释掉不用不用函数

#### Ctrl+Shift+C

31 32 33	/****Printer and Bluetooth initialization*****/	
34	<pre>gpio_led_init(); /*LED2 initialization*/</pre>	
35	<pre>my hsuart init(); /*Serial port initialization*</pre>	/
36	<pre>// printer_init(); /*printer initialization*/</pre>	
37		
38	/****Other routines Related initialization*****/	
39	<pre>// gpio_input_init();</pre>	
40	<pre>// Charge_init();</pre>	
41	<pre>// my_dac_init(DAC_L   DAC_R);</pre>	
42	<pre>// IIC_AT24C01_init();</pre>	
43	<pre>// power_sleep_wake_init();</pre>	
44	<pre>// rtc_calendar_init();</pre>	
45	<pre>// timer3_cap_init(66666);</pre>	
46	// timer3_init(500000 - 1); // 500ms, f=1MHz	

7、打开注释文件

#### Ctrl+Shift+X

44	ATHOTAGE CIMEL DAWNIN	
22	<pre>#include "uart_transfer.h"</pre>	
23	<pre>#include "wdt.h"</pre>	
24	/**************************************	*****
25		
26		
27	//Start Main function	
28	int main(void)	
29	- (	
30	bsp sys init();	/*System initialization*/
31		
32	/****Printer and Bluetooth ini	tialization******/
33		
34	<pre>gpio led init();</pre>	/*LED2 initialization*/
35	my hsuart init();	<pre>/*Serial port initialization*/</pre>
36	<pre>printer init();</pre>	<pre>/*printer initialization*/</pre>
37	void printer init (void)	
38	/****Other routines Related ini	tialization*****/
39	<pre>// gpio input init();</pre>	
40	<pre>// Charge init();</pre>	
41	// my dac init (DAC L   DAC R	.);
42	<pre>// IIC AT24C01 init();</pre>	



# 7 版本历史

日期	版本	变更历史
2022.11	1.0	新建
2022.12	2.0	增加"例程展示"和"常用设置"

表格 1 文件版本历史



# 声明

本手册由珠海极海半导体有限公司(以下简称"极海")制订并发布,所列内容均受商标、著作权、软件著作权相关法律法规保护,极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册, 一旦使用产品则表明您(以下称"用户")已知悉并接受本手册的所有内容。用户必须按照相关法律法规和 本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用,未经极海许可,任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有"◎"或"™"的"极海"或"Geehy"字样或图形均为极海的商标,其他在极海产品 上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用 户。

如果本手册中涉及任何第三方的产品、服务或知识产权,不应被视为极海授权用户使用前述第三方产品、服务或知识产权,除非在极海销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的,应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得,但本手册相关数据难免会出现 校正笔误或因测试环境差异所导致的误差,因此用户应当理解,极海对本手册中可能出现的该等错误无需承 担任何责任。本手册相关数据仅用于指导用户作为性能参数参照,不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品,并对极海产品的应用适用性进行有效验证和测试,以确认极 海产品满足用户自身的需求、相应标准、安全或其它可靠性要求;若因用户未充分对极海产品进行有效验证 和测试而致使用户损失的,极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时,应遵守当地所适用的所有法律法规。用户应了解产品可能受



到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制,用户(代表其本身、子公司及关联企业)应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海"按原样"(as is)提供,在适用法律所允许的范围内,极海不提供任何形式的明示或暗示担保,包括但不限于对产品适销性和特定用途适用性的担保。

对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷,极海概不承担责任。

7、责任限制

在任何情况下,除非适用法律要求或书面同意,否则极海和/或以"按原样"形式提供本手册的任何第 三方均不承担损害赔偿责任,包括任何一般、特殊因使用或无法使用本手册相关信息而产生的直接、间接或 附带损害(包括但不限于数据丢失或数据不准确,或用户或第三方遭受的损失)。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2022 珠海极海半导体有限公司 – 保留所有权利