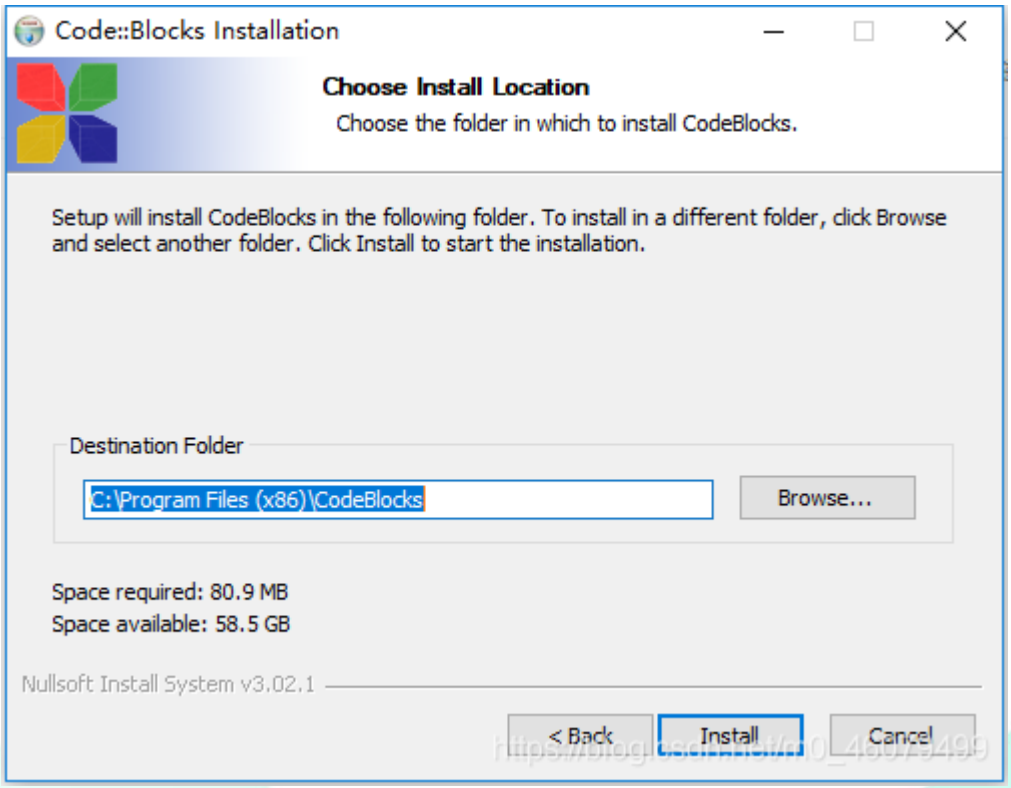
GW3323 开发说明

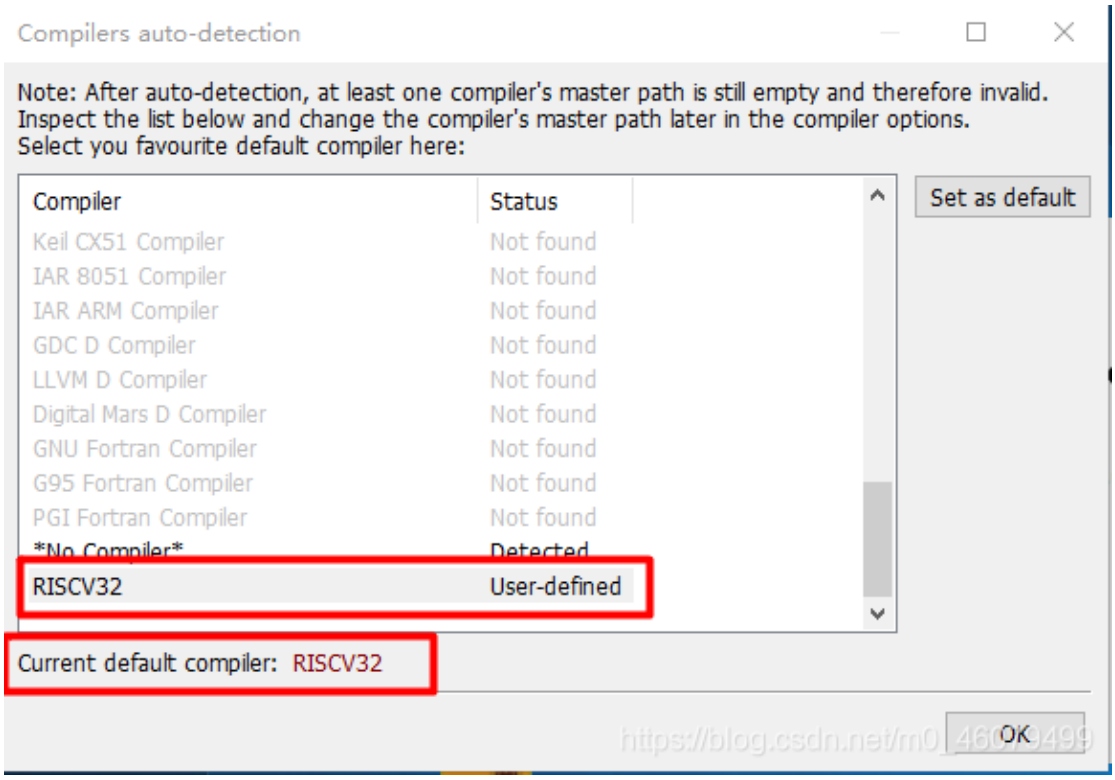
一、 编译环境说明

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)

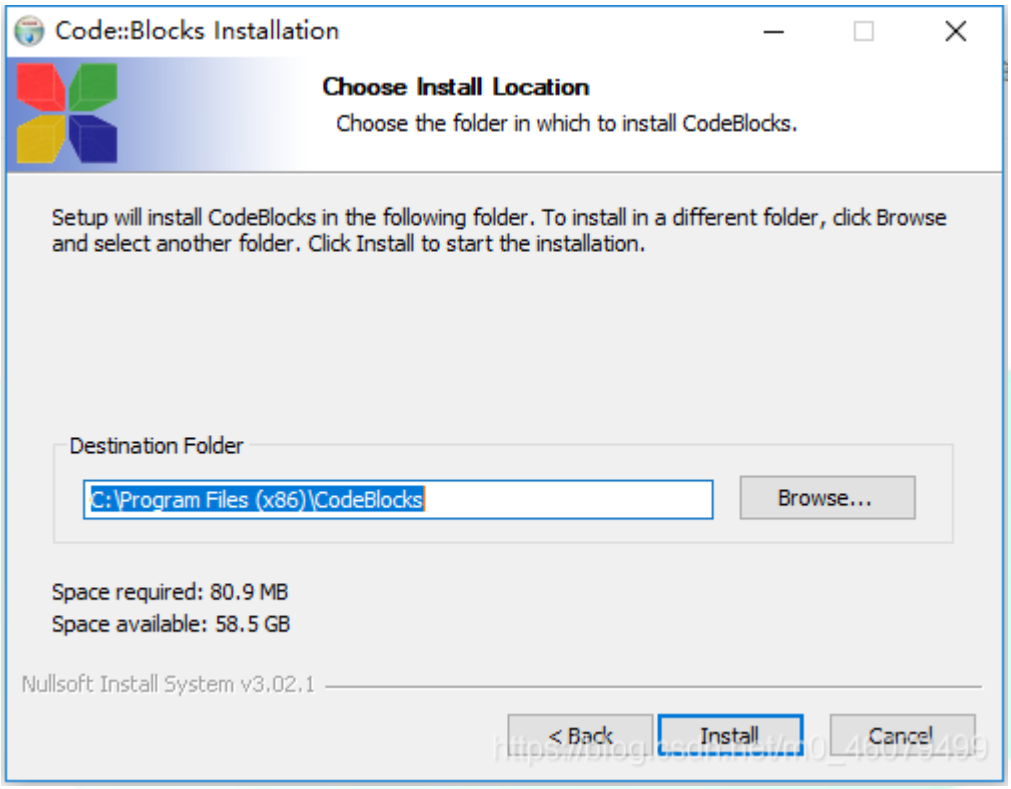
在开发中，我们一般是使用**串口打印**或者 GPIO 口和逻辑分析仪进行调试，**暂不支持断点调试和仿真。**

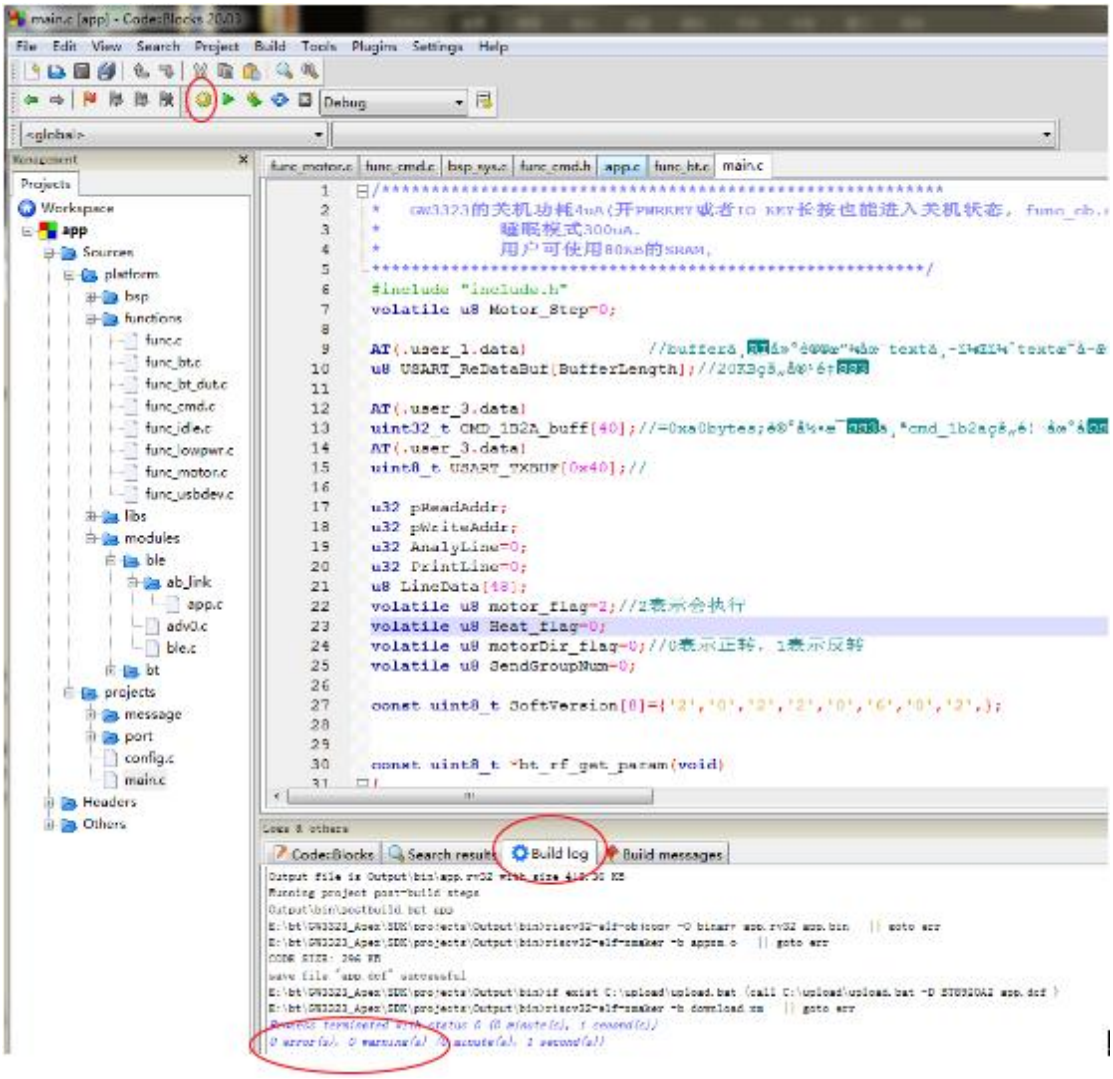
1、CodeBlock(IDE): (codeblocks-17.12)代码编辑器, 编译链接时会调用到 ToolChain 中提供的工具. 最终生成**烧写用的 dcf 文件**.





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



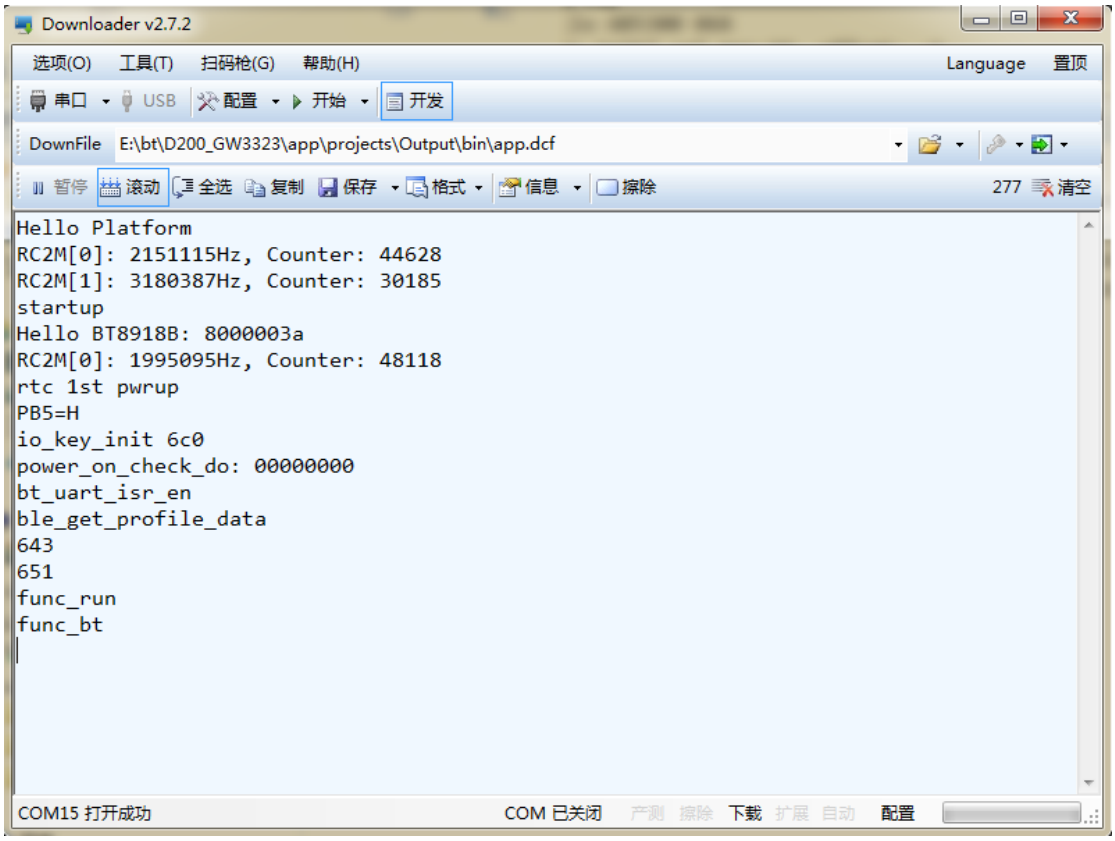


3、ToolChain: (RV32-Toolchain-Setup_vxxx)包含 RISC-V 编译器, Bin 文件转换工具等.

RV32-Toolchain-Setup.exe	2021/12/30 20:48	应用程序	24,977 KB

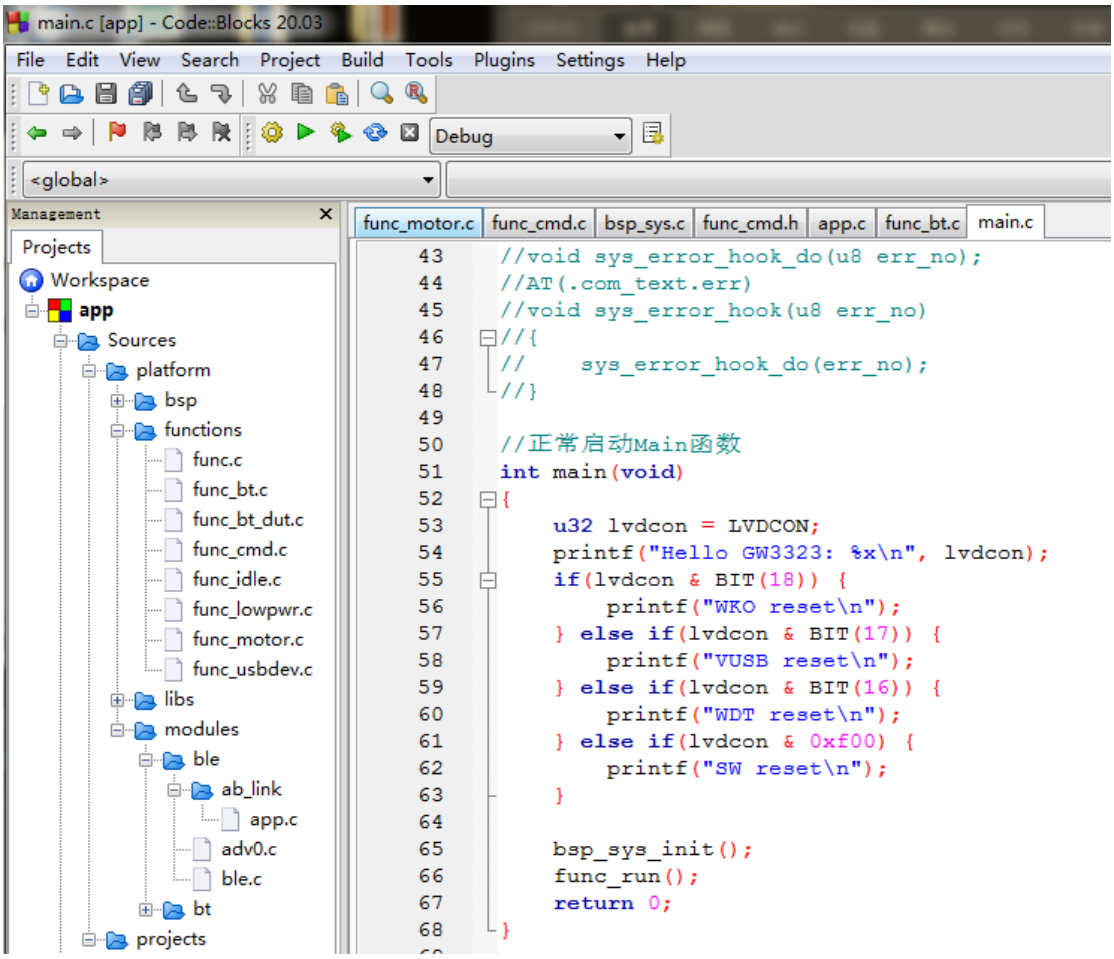
4、downloader 是烧录工具软件兼顾串口打印的功能，可以供开发人员调试，CP210x_Windows_Drivers 是 Xlink 烧录器的驱动程序。

CP210x_Windows_Drivers.rar	2022/3/11 17:08	WinRAR 压缩文件	3,656 KB
Downloader_v2.7.2.zip	2022/3/11 16:38	WinRAR ZIP 压缩...	2,488 KB



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.

二、 工程的介绍



先进行初始化，重点处理在 **func_enter();**

func_motor.c | func_cmd.c | bsp_sys.c | func_cmd.h | app.c | func_bt.c | main.c | bsp_sys.h | *func.c | func_lowpwr.h

```
105  void func_run(void)
106  {
107      printf("%s\n", __func__);
108      func_bt_chk_off();
109
110      while (1) {
111          func_enter();
112          switch (func_cb.sta) {
113  #if FUNC_BT_EN
114          case FUNC_BT:
115              func_bt();
116              break;
117  #endif
118
119  #if FUNC_IDLE_EN
120          case FUNC_IDLE:
121              func_idle();
122              break;
123  #endif // FUNC_IDLE_EN
124
125  #if FUNC_BT_DUT_EN
126          case FUNC_BT_DUT:
127              func_bt_dut();
128              break;
129
130          case FUNC_BT_FCC:
131              func_bt_fcc();
132              break;
133  #endif // IODM_TEST_MODE
134
135          case FUNC_PWROFF:
136              func_pwroff(1);
137              break;
138
139          default:
140              func_exit();
141              break;
142          }
143      }
144  }
```

func_motor.c | func_cmd.c | bsp_sys.c | func_cmd.h | app.c | func_bt.c | main.c | bsp_sys.h | *func.c | func_lowpwr.h

```
252
253     AT(.text.func.bt)
254     void func_bt(void)
255     {
256         uint16_t i;//,j;
257         uint8_t data;
258         uint8_t x_offset;
259     //    u16 sPrint_Dot=0;
260     //    u16 ccd=0;//
261         u16 Print_Dot=0;
262     //    u16 GroupOffset=0;
263         u8 GroupNum=0;
264     //    u16 PaperAdcRange;//前后差值
265     //    u16 PaperAdcPer;//上一刻的采样值
266         uint8_t Group_printbuf[5][MaxWide];//[0]96点以内;[1]~[2]192点以内
267         uint16_t Group_HTbuf[5];
268         printf("%s\n", __func__);
269
270         func_bt_enter();
271
272         while (func_cb.sta == FUNC_BT) {
273             func_bt_process();//喂狗，执行SPP或BLE程序，显示蓝牙连接状态
274             WDT_CLR();
275
276             if ( (PrintBuf.PrintTask) )//1点行已处理完
277             {
390             else  if( ((TMR3CON&BIT(0))==0) )//if( (motor_flag==2) )//有打
391             {//打印全部执行完成才进入此模块
506
507             func_bt_message(msg_dequeue());
508             func_bt_display();
509         }
510         printf("func_bt_exit!\n");
511         func_bt_exit();
512     }
513
```

关于定时器的操作在 **bsp_sys.c** 里面

func_motor.c | func_cmd.c | bsp_sys.c | func_cmd.h | app.c | func_bt.c | main.c | bsp_sys.h | *func.c | func_lowpwr.h

```
374   AT(.com_text.isr)
375   void timer3_init(void)
376   {
377       TMR3CON =  BIT(7);                  //Timer overflow interrupt enable
378       TMR3CNT = 0;
379       TMR3PR  = MotorSpeed_Init - 1;          //4步1点行，马达走纸速度达15.6mm/s;
380       TMR3CON |= BIT(2) ;          //Timer works in Counter Mode
381       sys_irq_init(IRQ_TMR3_VECTOR, 1, timer3_isr);
382   }
383
384   #if (TimerPrint == 1)
385   AT(.com_text.isr)
386   void timer4_isr(void)//
387   {
388   }
389   #elif (TimerPrint == 2)
390   AT(.com_rodata.isr)
391   const char str_t4[] = "&\n";//此数据不放在flash中，只能从内存里读取数据，load flash 会复位
392
393   AT(.com_text.isr)
394   FIQ void timer4_isr(void)//快速中断请求，
395   {
396           TMR4CPND = BIT(16);              //Clear Pending
397   //        printk(str_t4);
398           TPH_STB_RESET;
399           TMR4CON &= ~BIT(0);
400   }
401   #endif
402   AT(.com_text.isr)
403   void timer4_init(void)
404   {
405       TMR4CON =  BIT(7);                  //Timer overflow interrupt enable
406       TMR4CNT = 0;
407       #if (TimerPrint == 1)
408       TMR4PR  = 800  - 1;          //500ms, select xosc26_div 1M clk
409       TMR4CON |= BIT(2)| BIT(0);          //Timer works in Counter Mode
410
411       #elif (TimerPrint == 2)
412       TMR4PR  = 1600  - 1;           //25℃时加热1.02ms显影
413       TMR4CON |= BIT(2);          //Timer works in Counter Mode
414       sys_irq_init(IRQ_TMR4_VECTOR, 1, timer4_isr);
415       #endif
416   }
```

关于蓝牙的控制在 **spp.c** 和 **app.c**

func_motor.c | func_cmd.c | bsp_sys.c | func_cmd.h | app.c | func_bt.c | main.c | bsp_sys.h | *func.c | func_lowpwr.h | ble.c | spp.c

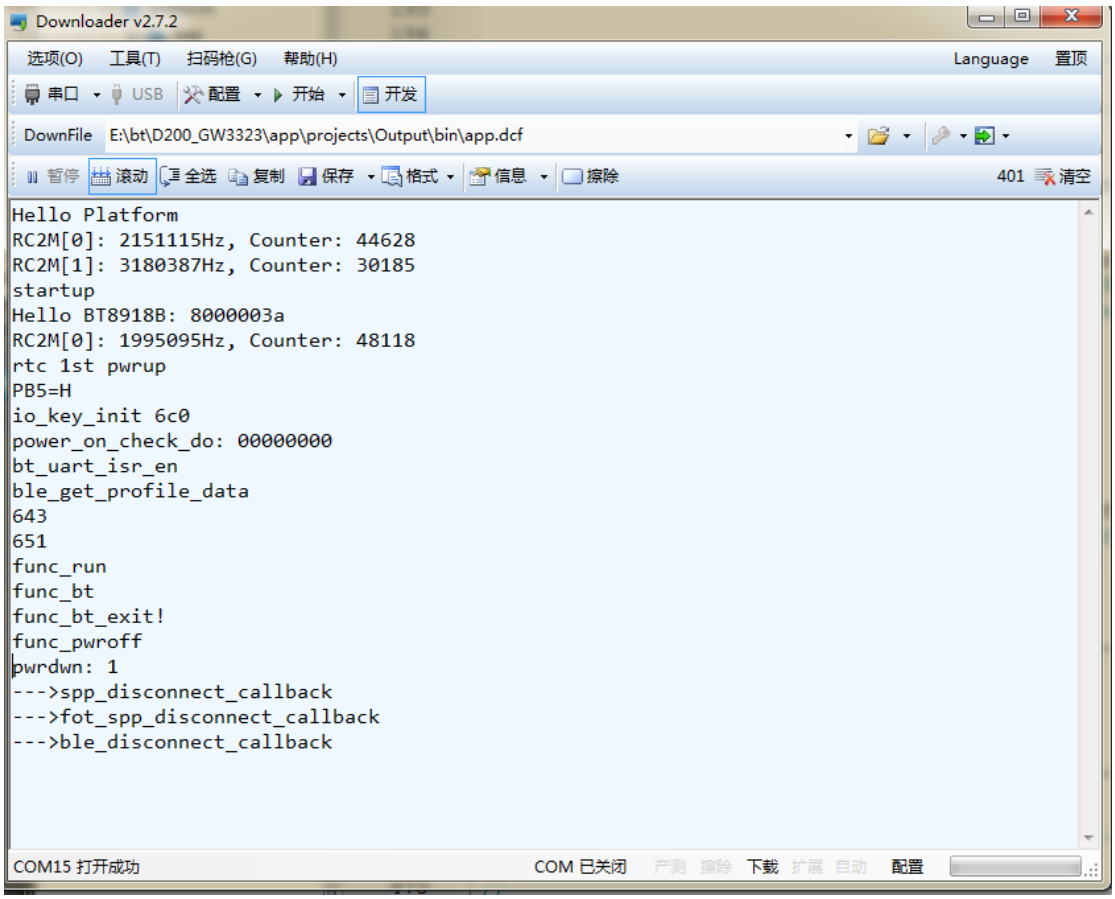
```
78  void spp_connect_callback(void)
79  {
80      printf("--->spp_connect_callback\n");
81  #if BT_SPP_FOT_EN
82      fot_spp_connect_callback();
83  #endif // BT_SPP_FOT_EN
84  }
85
86  void spp_disconnect_callback(void)
87  {
88      printf("--->spp_disconnect_callback\n");
89  #if BT_SPP_FOT_EN
90      fot_spp_disconnect_callback();
91  #endif // BT_SPP_FOT_EN
92  }
93
94  void spp_rx_callback(uint8_t *packet, uint16_t size)
95  {
96  #if BT_SPP_FOT_EN
97      if(fot_app_connect_auth(packet, size)){
98          fot_recv_proc(packet,size);
99          return;
100     }
101 #endif // BT_SPP_FOT_EN
102
103    uint16_t i;
104     uint8_t USARTReDataTemp;
105
106     printf("[%x-%x] (%x)=%x,%x,%x,%x,%x,%x,%x...%x,%x,%x\n'
107     for(i=0;i<size;i++)
108     {
109         USARTReDataTemp = *(packet+i);
110         USART_ReDataBuf[pReadAddr]= USARTReDataTemp;
111 //          printf(" [%x]:%x ", pReadAddr,USART_ReDataBuf[pRe
```

func_motor.c | func_cmd.c | bsp_sys.c | func_cmd.h | app.c | func_bt.c | main.c | bsp_sys.h | *func.c | func_lowpwr.h | ble.c | spp.c

```
189         .type = BLE_GATTS_UUID_TYPE_16BIT,
190         .uuid = app_notify_uuid16,
191     };
192     static gatts_service_base_st gatts_app_notify_base;
193
194
195     //-----------------------------------------------------------------------
196     // APP
197     //-----------------------------------------------------------------------
198     bool ble_send_packet(u8 *buf, u8 len)
199     {
200         return ble_tx_notify(gatts_tests_base.att_index, buf, len);
201     }
202     //ble每次收0x8c个字节，即140个。
203     static uint8_t gatt_callback_app(u8 *ptr, u8 len)
204     {//cmd_1d0e(退纸并打印);cmd_1b2a...;cmd_1d0c(进纸到撕纸位置)
205     //printf("BLE[%x]:%x  %x  %x  %x  %x  %x  %x  %x  %x  %x  \n",len,*p
206        uint16_t t,j,Line24_oft;
207         uint8_t Rec_No,USARTReDataTemp;
208     //  uint16_t Else_Len;
209
210         for(Rec_No=0;Rec_No<len;Rec_No++)
211         {
212             USARTReDataTemp = *(ptr+Rec_No);
213             USART_ReDataBuf[pReadAddr]= USARTReDataTemp;
214      //       printf(" [%x]:%x ",pReadAddr,USART_ReDataBuf[pReadAddr]);
215             pReadAddr = (pReadAddr+1)%BufferLength;
216             USART_ReceiveCNT++;
217              switch(USART_ReState)
218             {
424         return false;
425     }
426
```

三、 下载说明

用 **USB** 转串口的 **RX** 连接主控的 **PB3** 引脚，打开工程的**.dcf** 文件，配置好所需的功能，点击“开始”，即可看到如下下载成功并运行的程序。



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。