MBL 系列评估套件用户手册

新一代封装兼容型 Sub-1G 无线模块

E30-900MBL-01

成都亿佰特电子科技有限公司
Chengdu Ebyte Electronic Technology Co.,Ltd.

免责申明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。 文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵 权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责 任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反 言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中所得测试数据均为亿佰特实验室测试所得，实际结果可能略有差异。

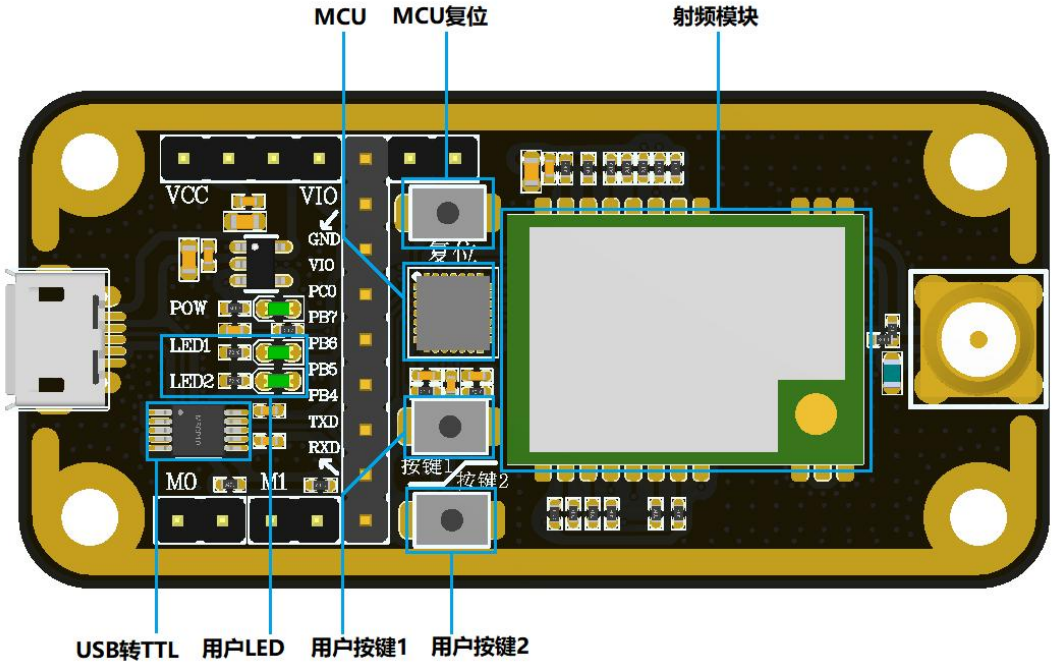
文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

最终解释权归成都亿佰特电子科技有限公司所有。

注 意 ：

由于产品版本升级或其他原因，本手册内容有可能变更。亿佰特电子科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，成都亿佰特电子科技有限公司尽全力在本手册中提供准确的信息，但是成都亿佰特电子科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

第一章 产品概述

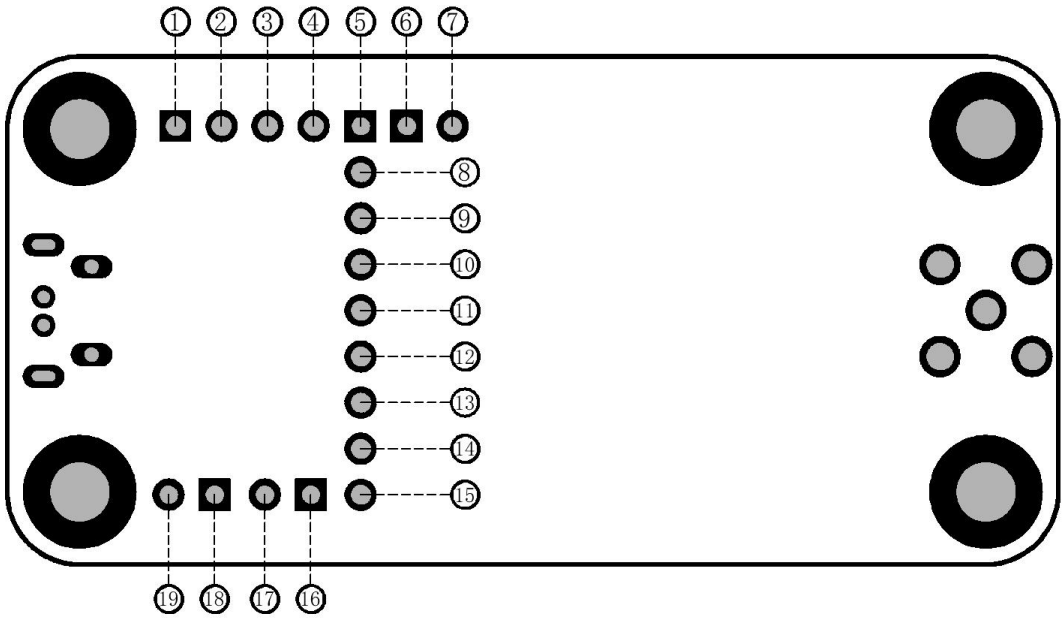


1.1 产品简介

MBL 系列评估套件旨在帮助用户快速评估亿佰特新一代封装兼容型无线模块。板上大部分管脚均已引出至两侧排针，开发人员可以根据实际需求，轻松通过跳线连接多种外围设备。

该套件提供完整的软件应用示例，助力客户快速上手无线数据通信开发。根据客户需求可以板载不同类型的 Sub-1G 无线模组。已支持的模组都具有引脚兼容的封装，可快速替换。

1.2 尺寸、接口描述



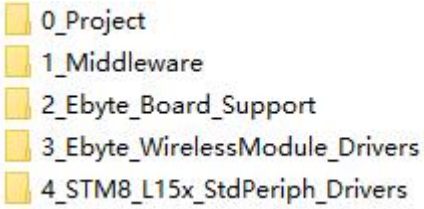
引脚序号	定义	功能说明
1	VCC	模块供电引脚，需要与 2 号脚短接给模块供电
2	3.3V	3.3V 电引出脚
3	3.3V	3.3V 电引出脚
4	VIO	MCU 供电引脚，需要与 3 号脚短接给 MCU 供电
5	GND	底板参考地
6	REST	MCU 外部复位引脚
7	SWIM	MCU 的 SWIM 引脚
8	VIO	MCU 供电引脚
9	PC0	模块复位引脚
10	PB7	模块 MISO 引脚
11	PB6	模块 MOSI 引脚
12	PB5	模块 SCLK 引脚
13	PB4	模块 NSS 引脚
14	TXD	MCU 串口 TXD
15	RXD	MCU 串口 RXD
16	M1	模块模式切换引脚（详见模块产品手册）
17	GND	底板参考地
18	M0	模块模式切换引脚（详见模块产品手册）
19	GND	底板参考地

1.3 支持列表

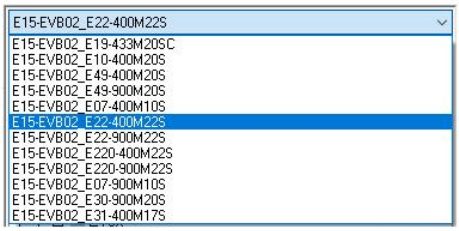
	射频方案	制造商	模块型号
1	CC1101	Texas Instruments	E07-400M10S
2	CC1101	Texas Instruments	E07-900M10S
3	SI4438	Silicon Labs	E30-400M20S
4	SI4463	Silicon Labs	E30-900M20S
5	SX1278	Semtech	E32-400M20S
6	SX1276	Semtech	E32-900M20S

第二章 软件简介

2.1 目录结构

	事项	说明
1	文件目录	从官网可以下载到示例工程，打开目录如下图所示 
2	目录说明	可以使用 IAR For STM8 开发环境找到入口文件打开工程 **E15-EVB02 Demo** //主文件夹 **0_Project** IAR_for_Stm8 //工程文件夹 使用 IAR 打开工程 **1_Middleware** Kfifo //通用数据队列 Produce //PC测试 **2_Ebyte_Board_Support** E15-EVB02 //板载资源初始化 **3_Ebyte_WirelessModule_Drivers** E07xMx //E07模块驱动 E10xMx //E10模块驱动 E19xMx //E19模块驱动 E22xMx //E22模块驱动 E30xMx //E30模块驱动 E31xMx //E31模块驱动 E49xMx //E49模块驱动 E220xMx //E220模块驱动 **4_STM8_L15x_StdPeriph_Drivers**

2.2 IAR 工程

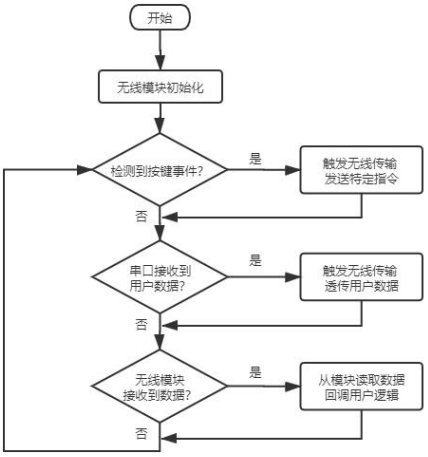
	事项	说明
	工程结构	使用 IAR For STM8 开发环境打开工程可以看到基本结构 
	切换工作空间	C/C++ Compiler 选项中定义了全局宏定义与文件路径，用来区别不同模块的驱动文件。当切换工作空间时，将会使用不同的宏定义，从而切换不同模块的驱动文件  改变了 Drivers/Ebyte/RF 的 Exclude from build 属性，即选择目标模块驱动文件夹参与编译过程。改变了工程 C/C++ Compiler 中的 Additional include，即指定模块驱动文件路径。改变了工程 C/C++ Compiler 中的 Defined symbols，即定义了全局宏定义，帮助配置模块驱动属性。 

2.3 主函数

main.c 中即为主函数入口。演示功能流程简化如下过程：

	事项	说明
1	按键功能	有按键按下那就无线发送指令数据。实质上即为发送特定字符串“ping" 并期望接收到回应“pong”
2	串口数据转无线发送	串口收到数据后自动开始无线透传数据，当然其中包含有部分特殊指令响应，主要用于特殊测试，用户可以忽略。发送完成后会自动回调用户函数，从而自行处理发送逻辑。
3	无线接收数据	一般都是读取模块内部状态标识来判断是否有数据，底层驱动会拷贝数据并传递给用户回调函数，从而自行处理接收逻辑

软件流程简化如下图所示：

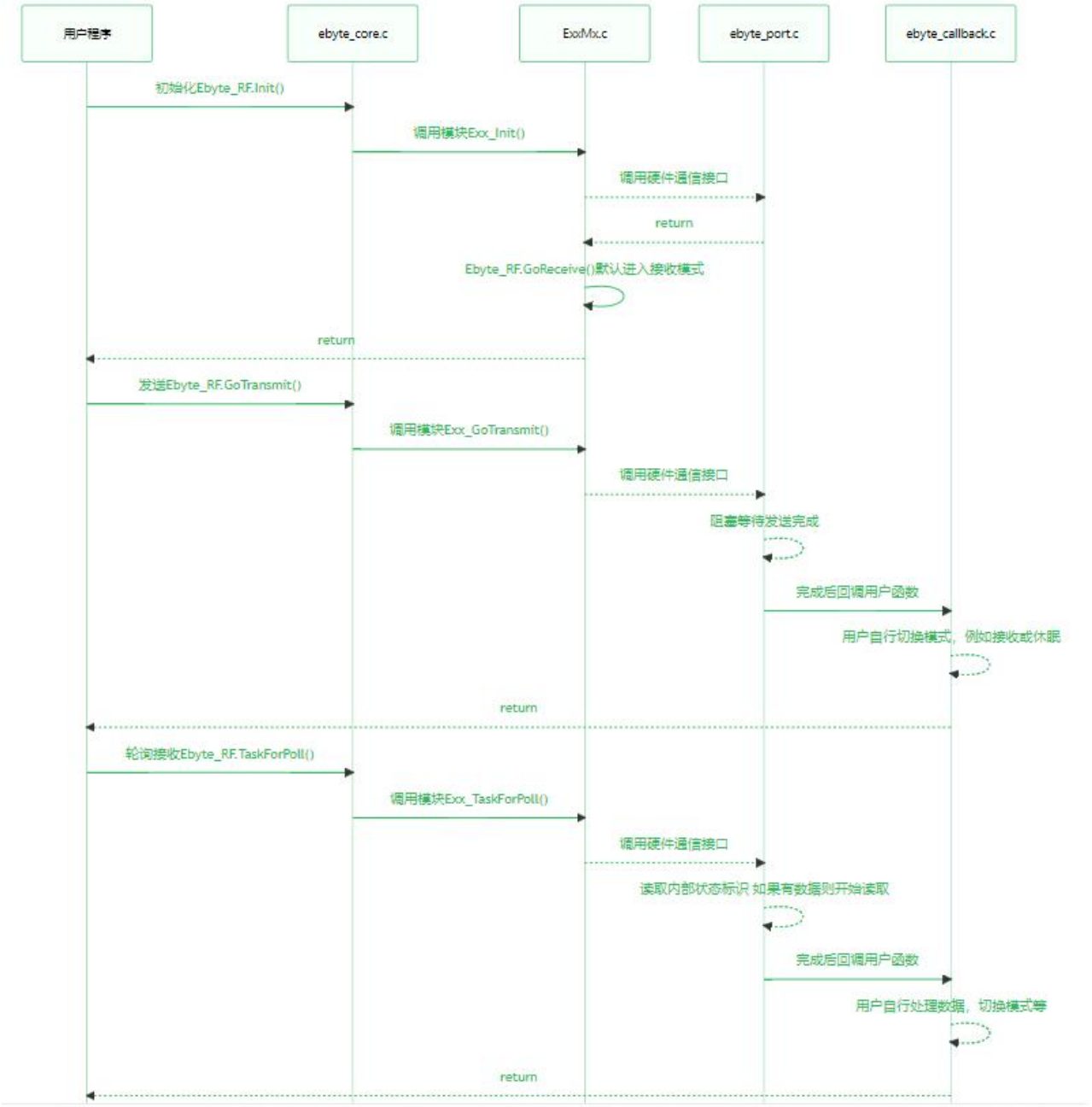


2.4 收发时序

无线模块存在多种运行状态，只能在相应的状态完成特定的功能。从最简单的收发数据来说，只考虑发送模式、接收模式。

	事项	说明
1	接收模式	默认初始化完成后自动进入接收模式。实质上即为初始化中调用了接收函数从而进入了接收模式。如需考虑初始化完成后进入其他模式，例如休眠，直接用同类型函数 Go_xxxxx() 替换即可。
2	发送模式	调用发送函数时，底层驱动实际上先将模块切换进入待机模式，通常在此模式下完成调制参数配置，例如频率、功率、频偏等等。在参数配置正确后，逐步进入一些中间模式，开启内部 FIFO、PA、外部 XTAL 等，电流消耗也逐步攀升。最终切换进入发送模式，触发无线数据传输。完成后模块进入待机模式，此状态下无法继续收发，需要用户在回调函数中自行处理下一步模式。当功能复杂后，需要连续接收或者连续发送，请根据芯片特性进一步切换其他模式。

时序图如下所示：



2.5 程序设计

	文件	关键说明
1	ebyte_core.h	定义了一个模块结构体，抽象出基本功能，底层模块的函数将绑定到该结构体。当用于简单收发应用时，没必要去了解每个模块的底层工作细节，直接调用抽象出的函数就可以开始收发数据。如需自定义一些功能，也可以考虑集成到该结构体内。如果对底层模块的功能函数足够了解，也可以直接去掉 ebyte_core.c/h 文件，分层之间并没有强耦合。

		typedef struct
		{
		uint8_t (*Init)(void); //初始化
		uint8_t (*GoTransmit)(uint8_t *buffer, uint8_t size); //切换发送模式 开始传输数据
		uint8_t (*GoSleep)(void); //切换到休眠模式 低功耗用到
		uint8_t (*GoReceive)(void); //切换到接收模式 开始监听数据
		uint8_t (*TaskForPoll)(void); //轮询函数 可以主循环周期调用 也可以视情况放入中断
		void (*TaskForIRQ)(void); //暂时保留，不必使用。用于将来扩展中断收/发
		uint8_t (*GetStatus)(void); //获取模块状态(软件状态机)
		uint8_t * (*GetName) (void); //获取模块识别码 为字符串 例如 "E22-400M22S"
		uint8_t (*GetDriver) (void); //获取软件版本号
		}Ebyte_RF_t;

2	ebyte_exx.c	是具体模块驱动文件，一般都是封装好的，不需要用户去改动，只需要考虑怎么把数据从这个"盒子"里输入输出。

3	ebyte_port.c	专门用来绑定不同硬件平台下的 SPI 和 GPIO，抽象为"盒子"的输入。用户需要将自己的硬件平台中的通信接口按照注释填充到固定的位置。一般来说，就是提供 SPI 的收发函数和引脚的电平控制。有些模块稍微特殊，例如 E49 采用半双工 SPI，要是懒得写通信驱动那就直接将 IO 绑定到固定位置，剩下的交给模块驱动自己模拟 IO 实现通信。如下图所示，在注释中要求提供 SPI 接口位置填入具体的收发函数，由 send 传入 SPI 发送数据，由 result 返回 SPI 接收数据。
		/*!
		* @brief 配置目标硬件平台SPI接口收发函数
		* @param send EBYTE驱动库上层调用需要传输的数据 1 Byte
		* @return SPI 接收的数据 1 Byte
		*/
		uint8_t Ebyte_Port_SpiTransmitAndReceivce(uint8_t send)
		{
		uint8_t result = 0;
		/* 必须提供 SPI接口 */
		result = Ebyte_BSP_SpiTransAndRecv(send); //用户填充函数
		return result;
		}

	ebyte_callback.c	专门用来绑定用户自己的收发逻辑，抽象为"盒子"的输出。本质上，模块驱动就是在确定发送或接收完成后直接调用用户的回调函数。如下图所示，在 To-do 提示位置填充用户的逻辑函数即可。state 由模块驱动传出，实际由 Exx_GoTransmit()函数处理，当功能复杂时可以考虑修改以支持更多情况。
		/*!
		* @brief 发送完成回调接口 由客户实现自己的发送完成逻辑
		* @param state 上层回调提供的状态码 客户请根据示例注释找到对应区域
		*/
		void Ebyte_Port_TransmitCallback(uint16_t state)
		{
		/* 发送 正常完成 */
		if(state &= 0x0001)
		{
		//To-do 实现自己的逻辑
		UserTransmitDoneCallback();
		}
		/* 发送 其他情况 */
		else
		{
		//To-do 实现自己的逻辑
		}
		}

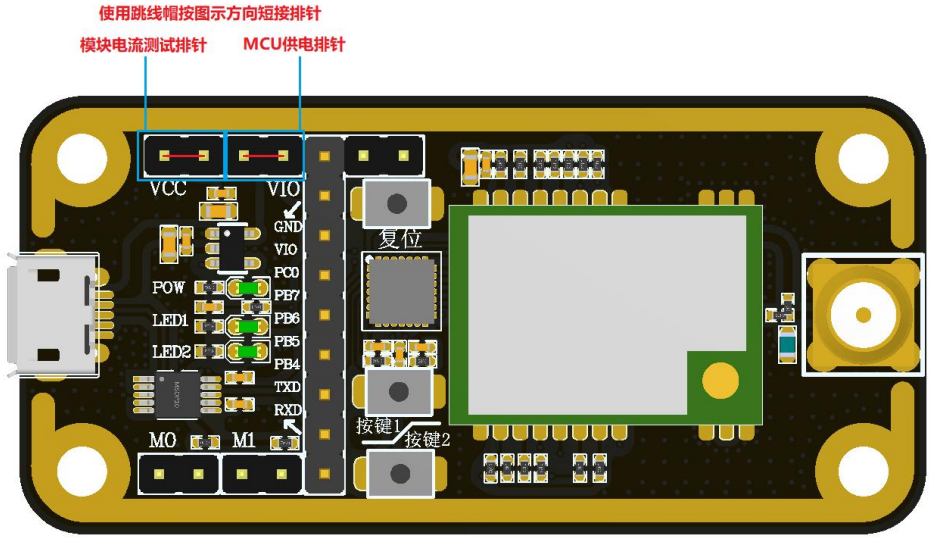
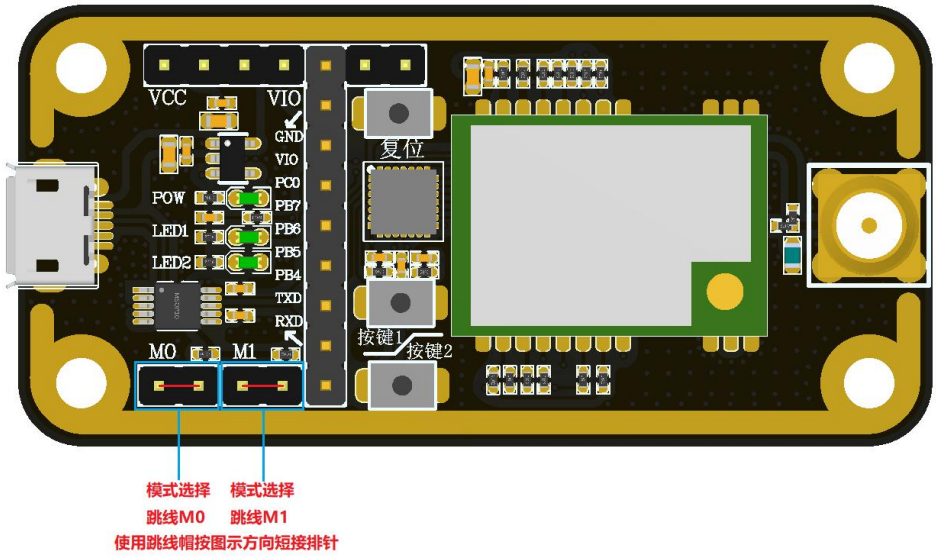
	ebyte_exx.h	定义了一些常规的调制参数，一般不需要修改，可以在其中自行进行调整。注意，修改时请理解注释中得说明，模块驱动中对参数有范围检查，错误的调制参数

		会导致初始化失败。如下图示例 FSK 调制参数：
		`#define E07_DATA_RATE 1200 //空速 1.2 KBps`
		`#define E07_FREQUENCY_DEVIATION 14300 //频偏 14.3 K`
		`#define E07_BANDWIDTH 58000 //接收频宽 58 K`
		`#define E07_OUTPUT_POWER 10 //功率 [10 7 5 0 -10 -15 -20 -30]`
		`#define E07_PREAMBLE_SIZE 4 //前导码长度 [0:2 1:3 2:4 3:6 4:8 5:12 6:16 7:24]`
		`#define E07_SYNC_WORD 0x2DD4 //同步字`
		`#define E07_IS_CRC 1 //CRC开关 [0:关闭 1:开启]`

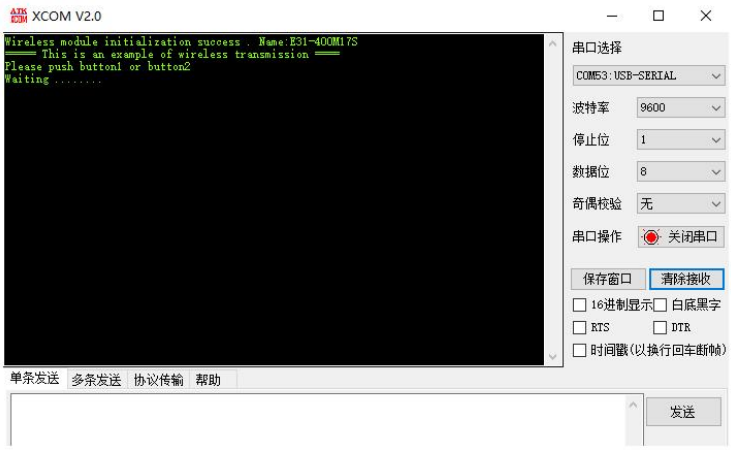
	board.c	STM8 外设初始化，涉及 SPI、TIMER、GPIO 等等，与所用硬件强耦合。
	board_button.c	按键事件队列，从数据结构来说就是个 FIFO。定时器检测到按键后会把对应事件存入队列等待主循环响应。
	board_mini_printf.c	简化的 printf，虽然功能缩水但占用体积很小。工程中的 DEBUG 宏主要依赖此文件提供的 mprintf。
	ebyte_kfifo.c	用于串口数据接收，优化过的通用 FIFO 队列，适合高速缓存。
	ebyte_debug.c	用于连接 PC 机进行一些测试，一般无需使用。
	stm8l15x_it.c	所有中断函数入口，将对于串口、定时器、按键 IO 等中断服务函数都集中在此

第三章 快速演示

3.1 信号线连接

	事项	说明
1	电源跳线帽	
2	模式选择跳线帽	
3	辅助	USB 线缆、天线、PC 等

3.2 串口助手

	事项	说明
1	设备管理器 查看串口编号	
2	串口软件	
3	按键通信示例	#RECV 标识符，仅用于提示，表示无线模块接收到的数据。 #SEND 标识符，仅用于提示，表示无线模块发送了的数据 
4	串口数据透传	串口数据透传 通过 XCOM 直接传输所需内容 

第四章 常见问题

4.1 传输距离不理想

- 当存在直线通信障碍时，通信距离会相应的衰减；
- 温度、湿度，同频干扰，会导致通信丢包率提高；
- 地面吸收、反射无线电波，靠近地面测试效果较差；
- 海水具有极强的吸收无线电波能力，故海边测试效果差；
- 天线附近有金属物体，或放置于金属壳内，信号衰减会非常严重；
- 功率寄存器设置错误、空中速率设置过高（空中速率越高，距离越近）；
- 室温下电源低压低于推荐值，电压越低发功率越小；
- 使用天线与模块匹配程度较差或天线本身品质问题。

4.2 模块易损坏

- 请检查供电电源，确保在推荐供电电压之间，如超过最大值会造成模块永久性损坏；
- 请检查电源稳定性，电压不能大幅频繁波动；
- 请确保安装使用过程防静电操作，高频器件静电敏感性；
- 请确保安装使用过程湿度不宜过高，部分元件为湿度敏感器件；
- 如果没有特殊需求不建议在过高、过低温度下使用。

4.3 误码率太高

- 附近有同频信号干扰，远离干扰源或者修改频率、信道避开干扰；
- 电源不理想也可能造成乱码，务必保证电源的可靠性；
- 延长线、馈线品质差或太长，也会造成误码率偏高。

修订历史

版本	修订日期	修订说明	维护人
1.0	2021-09-22	初始版本	JH
1.1	2022-12-29	修改模块示意图及使用方式	HWJ

关于我们

销售热线：4000-330-990 公司电话：028- 61543675
技术支持：support@cdebyte.com 官方网站：www.ebyte.com
公司地址：四川省成都市高新西区西区大道 199 号 B5 栋

成都亿佰特电子科技有限公司
Chengdu Ebyte Electronic Technology Co.,Ltd.