MBL Series Evaluation Kit User Manual

New generation package compatible Sub-1G wireless module

E07-900MBL-01

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

Contents

Disclaimer....2
1. Product Overview....3
....错误！未定义书签。
1.1 Brief Introduction....3
1.2 Size, interface description....4
....4
1.3 Support list....5
2. Software Description....5
2.1 Directory Structure....5
2.2 IAR Project....6
2.3 Main function....7
2.4 Transceiving timing....8
2.5 Programming....9
3 Quick demo....12
3.1 Signal line connection....12
3.2 Serial port assistant....12
4. Common problem....14
4.1 Unsatisfactory transmission distance....14
4.2 Module is easily damaged....14
4.3 Bit error rate is too high....14
Revise history....14
About us....15

Disclaimer

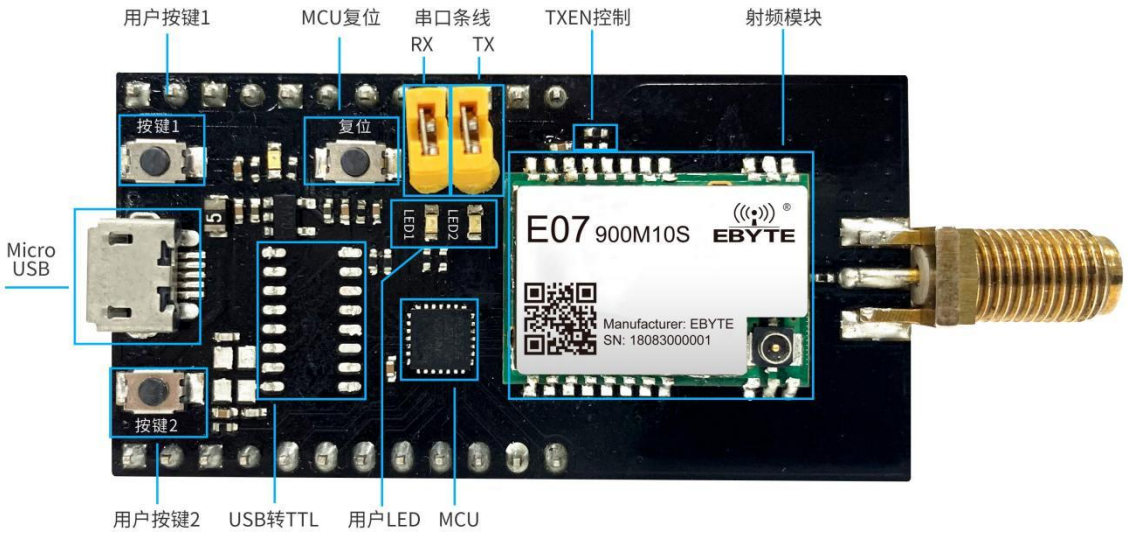
EBYTE reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. Reproduction, use, modification or disclosure to third parties of this document or any part thereof without the express permission of EBYTE is strictly prohibited.

The information contained herein is provided “as is” and EBYTE assumes no liability for the use of the information. No warranty, either express or implied, is given, including but not limited, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by EBYTE at any time. For most recent documents, visit www.ebyte.com.

Note:

The contents of this manual are subject to change due to product version upgrades or other reasons.Chengdu Ebyte Electronic Technology Co.,Ltd. reserves the right to make changes to the contents of this manual without notice or suggestion.This manual serves only as a user guide and Chengdu Ebyte Electronic Technology Co.,Ltd. endeavours to provide accurate information in this manual, but Chengdu Billionaire Electronics Co., Ltd. does not ensure that the contents are completely error-free and that all statements, information and suggestions in this manual do not constitute any express or implied warranty.

1. Product Overview

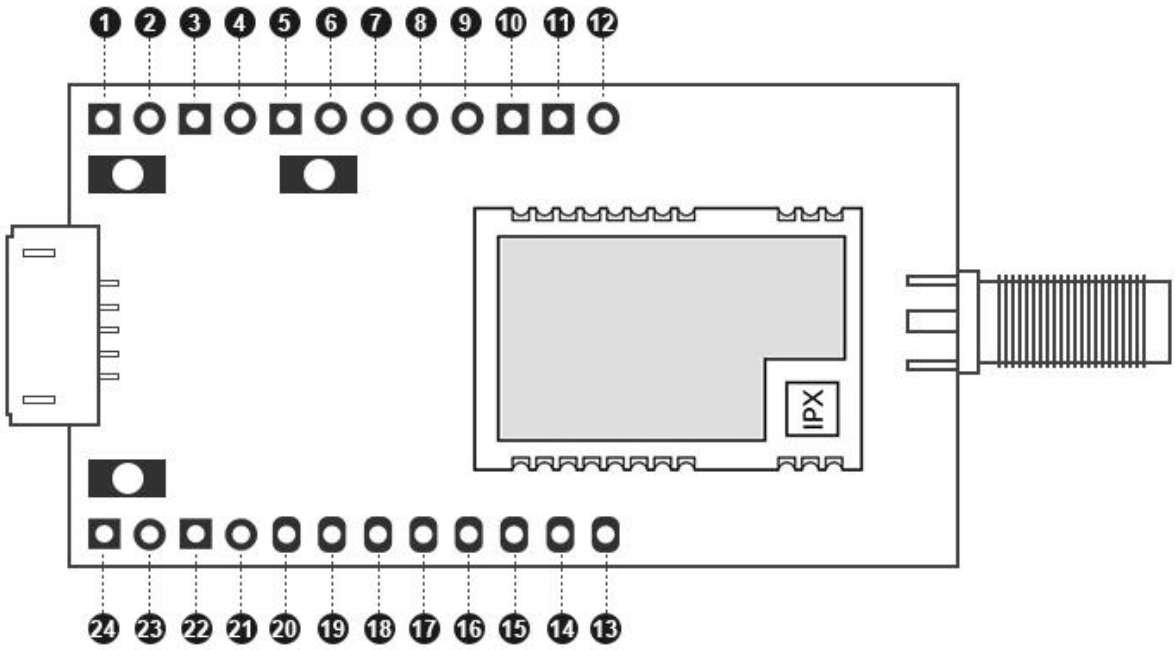


1.1 Brief Introduction

The MBL series evaluation kits are designed to help users quickly evaluate Ebyte's new generation package compatible wireless modules. Most of the pins on the board have been led out to the pin headers on both sides, developers can easily connect a variety of peripheral devices through jumpers according to actual needs.

The kit provides complete software application examples to help customers quickly get started with wireless data communication development. Different types of Sub-1G wireless modules can be mounted on-board according to customer needs. All supported modules have pin-compatible packages that can be quickly replaced.

1.2 Size, interface description



No.	Definition	Function Description
1	3.3V	3.3V Electric pin
2	3.3V	3.3V Electric pin
3	GND	Floor reference ground
4	GND	Floor reference ground
5	3.3V	3.3V Electric pin
6	GND	Floor reference ground
7	REST	MCUExternal reset pin
8	SWIM	SWIM pin of MCU
9	3.3V	3.3V Electric pin
10	VCC	Module power supply pin, need to be short-circuited with pin 9 to supply power to the module
11	PC6	MCU ordinary IO
12	PC5	MCU ordinary IO
13	PB3	MCU ordinary IO
14	PB2	MCU ordinary IO
15	PB1	MCU ordinary IO
16	PB0	MCU ordinary IO
17	M1	Module mode switch pin (see module product manual for details)
18	M0	Module mode switch pin (see module product manual for details)

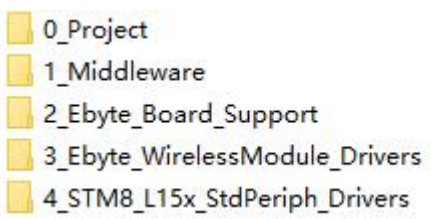
19	PD1	MCU ordinary IO
20	PD0	MCU ordinary IO
21	GND	Floor reference ground
22	GND	Floor reference ground
23	5V	5V Electric pin
24	5V	5V Electric pin

1.3 Support list

	RF Chip	Manufacturer	Module model
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	LLCC68	Semtech	E220-400M22S
6	LLCC68	Semtech	E220-900M22S
7	SX1278	Semtech	E32-400M20S
8	SX1276	Semtech	E32-900M20S
9	SX1268	Semtech	E22-400M22S
10	SX1262	Semtech	E22-900M22S

2. Software Description

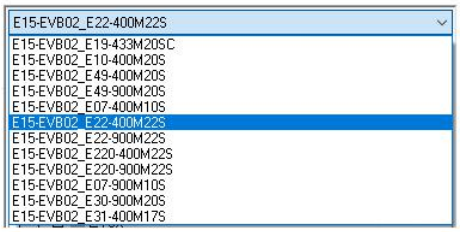
2.1 Directory Structure

	Matter	Explanation
1	File Directory	You can download the sample project from the official website, open the directory as shown in the figure below 
2	Catalog description	You can use the IAR For STM8 development environment to find the entry file to open the project

		├─ **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 Project

	Matter	Explanation
	Engineering structure	Use the IAR For STM8 development environment to open the project and you can see the basic structure

		
	Switch workspace	The global macro definition and file path are defined in the C/C++ Compiler option to distinguish the driver files of different modules. When switching workspaces, different macro definitions will be used to switch the driver files of different modules  Changed the Exclude from build property of Drivers/Ebyte/RF, that is, select the target module driver folder to participate in the compilation process. Changed the Additional include in the project C/C++ Compiler, that is, specify the module driver file path. The Defined symbols in the project C/C++ Compiler have been changed, that is, global macro definitions have been defined to help configure the module driver properties. 

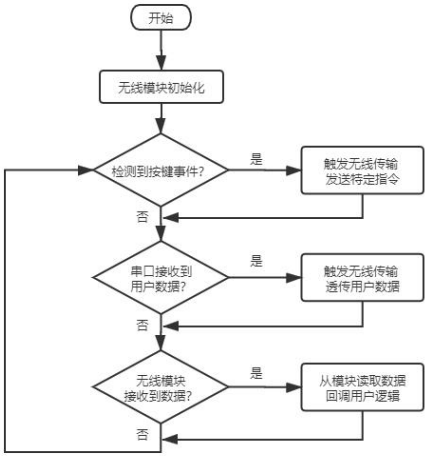
2.3 Main function

The main function entry is in main.c. The demonstration function process is simplified as follows:

	Item	Explanation
1	Key Function	If a button is pressed, the command data will be sent wirelessly. In essence, it means sending a specific string "ping" and expecting to receive a response "pong"

2	Serial data transfer to wireless transmission	After the serial port receives the data, it automatically starts to transmit the data wirelessly. Of course, it contains some special command responses, which are mainly used for special tests and can be ignored by the user. After the sending is completed, the user function will be automatically called back to handle the sending logic by itself.
3	Receive data wirelessly	Generally, the internal status identifier of the module is read to determine whether there is data, and the underlying driver will copy the data and pass it to the user callback function, so as to process the receiving logic by itself

The software process is simplified as shown in the figure below:

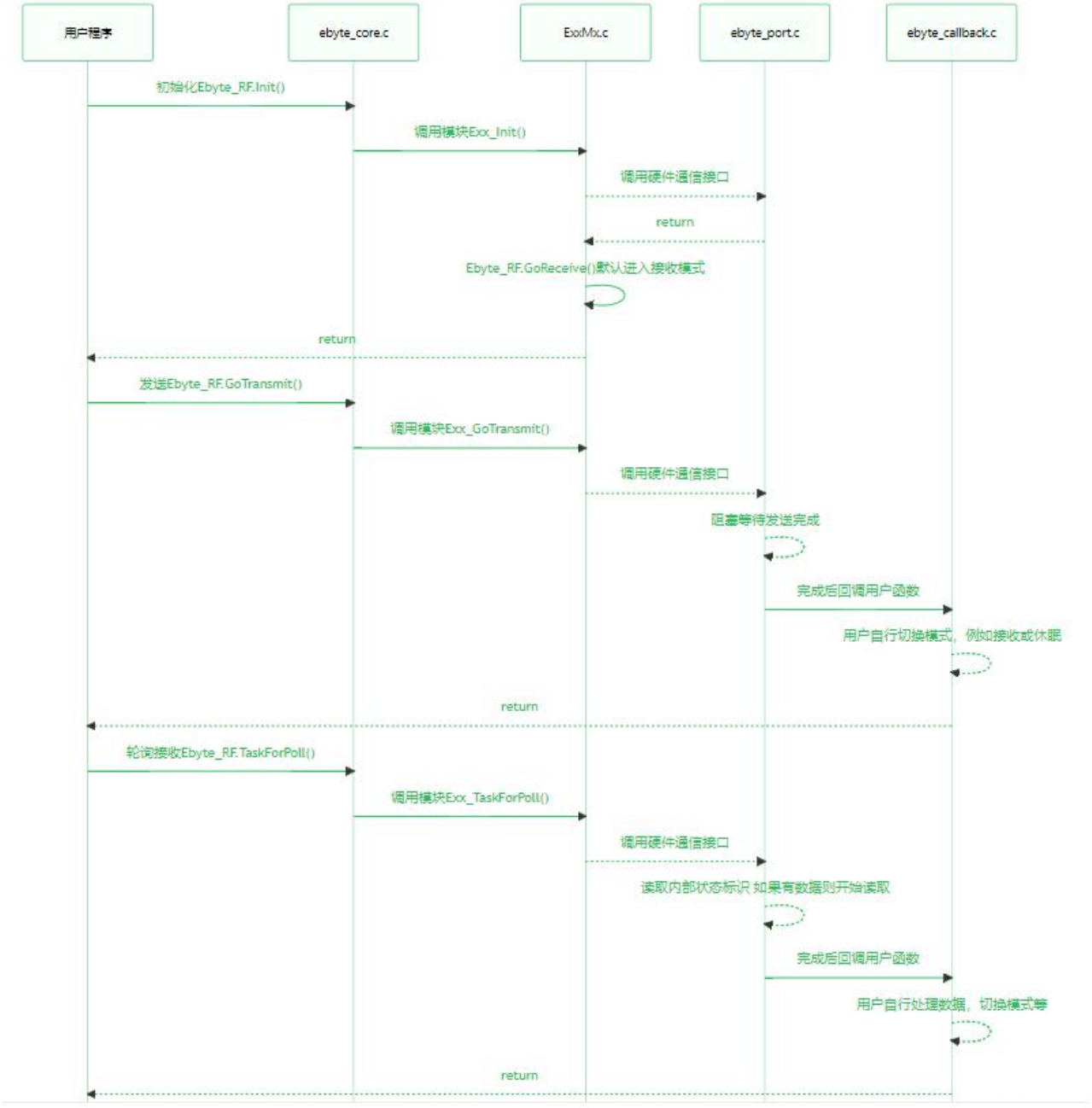


2.4 Transceiving timing

The wireless module has multiple operating states, and can only perform specific functions in the corresponding state. From the simplest way of sending and receiving data, only the sending mode and the receiving mode are considered.

	Item	Explanation
1	Receive mode	After the default initialization is completed, it will automatically enter the receiving mode. In essence, the receiving function is called during initialization, and the receiving mode is entered. If you need to consider entering other modes after initialization, such as sleep, just replace it with the same type of function Go_xxxxx().
2	Send mode	When calling the sending function, the bottom driver actually switches the module into the standby mode first, and usually completes the modulation parameter configuration in this mode, such as frequency, power, frequency offset, etc. After the parameter configuration is correct, gradually enter some intermediate modes, turn on the internal FIFO, PA, external XTAL, etc., and the current consumption also gradually rises. Finally, it switches to the sending mode and triggers wireless data transmission. After completion, the module enters the standby mode. In this state, it cannot continue to send and receive. The user needs to handle the next mode in the callback function. When the function is complicated, continuous reception or continuous transmission is required. Please switch to other modes according to the characteristics of the chip.

The timing diagram is as follows:



2.5 Programming

	File	Key note
1	ebyte_core.h	A module structure is defined to abstract the basic functions, and the functions of the underlying module will be bound to this structure. When used for simple sending and receiving applications, there is no need to understand the underlying working details of each module, and you can start sending and receiving data directly by calling the abstracted function. If you need to customize some functions, you can also consider integrating them into the structure. If you know enough about the functional functions of the underlying module, you can also directly remove the ebyte_core.c/h file, and there is

		no strong coupling between the layers 。 `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	It is a specific module driver file, which is usually packaged and does not need to be modified by the user. You only need to consider how to input and output data from this "box".
3	ebyte_port.c	It is specially used to bind SPI and GPIO under different hardware platforms, abstracted as "box" input. The user needs to fill the communication interface in his hardware platform to a fixed position according to the comment. Generally speaking, it is to provide the SPI transceiver function and pin level control. Some modules are slightly special. For example, E49 uses half-duplex SPI. If you are too lazy to write a communication driver, you can directly bind the IO to a fixed location, and leave the rest to the module to drive its own analog IO to achieve communication. As shown in the figure below, in the comments, it is required to provide the SPI interface location to fill in the specific transceiver function, send the SPI to send data, and the result to return the SPI to receive data. `/*!` `* @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	It is specially used to bind the user's own sending and receiving logic, abstracted as the output of the "box". Essentially, the module driver directly calls the user's callback function after confirming that the sending or receiving is completed. As shown in the figure below, just fill in the user's logic function at the To-do prompt. The state is transmitted by the module driver, and is actually processed by the Exx_GoTransmit() function. When the function is complicated, you can consider modifying it to support

		more situations. ```/*! * @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(uint16e_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	Some conventional modulation parameters are defined, which generally do not need to be modified, and can be adjusted by themselves. Note that when modifying, please understand the explanation in the comment. There is a range check for the parameters in the module driver, and the wrong modulation parameter will cause the initialization to fail. The following figure shows FSK modulation parameters: ```#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 peripheral initialization, involving SPI, TIMER, GPIO, etc., is strongly coupled with the hardware used.
	board_button.c	The key event queue is a FIFO in terms of data structure. After the timer detects the button, it will store the corresponding event in the queue and wait for the main loop to respond.
	board_mini_printf.c	Simplified printf, although the function is reduced, but it takes up a small volume. The DEBUG macro in the project mainly depends on the mprintf provided by this file.
	ebyte_kfifo.c	Used for serial port data reception, optimized general-purpose FIFO queue, suitable for high-speed cache.
	ebyte_debug.c	It is used to connect to a PC for some tests and generally does not need to be used.
	stm8l15x_it.c	All interrupt function entrances will focus on the interrupt service functions such as serial port, timer, button IO, etc.

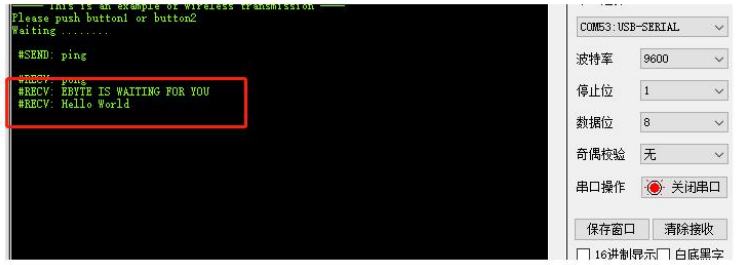
3 Quick demo

3.1 Signal line connection

	File	Key note
1	Serial jumper cap	
2	RF module jumper cap	
3	Assist	USB cable, PC, etc.

3.2 Serial port assistant

	Matter	Explanation
1	Device manager View serial port number	

2	Serial port software	
3	Push button communication example	#RECV Identifier, only used as a reminder, indicates the data received by the wireless module. #SEND Identifier, only for reminding, indicating the data sent by the wireless module 
4	Serial data transmission	Serial data transmission through XCOM direct transmission of required content 

4. Common problem

4.1 Unsatisfactory transmission distance

- When there is a straight line communication obstacle, the communication distance will be attenuated accordingly;
- Temperature, humidity, and co-frequency interference will increase the communication packet loss rate;
- The ground absorbs and reflects radio waves, and the test effect is poor when it is close to the ground;
- Sea water has a strong ability to absorb radio waves, so the seaside test effect is poor;
- If there is a metal object near the antenna or placed in a metal shell, the signal attenuation will be very serious;
- The power register is set incorrectly, and the air speed is set too high (the higher the air speed, the closer the distance);
- The low voltage of the power supply at room temperature is lower than the recommended value, and the lower the voltage, the lower the power output;
- The matching degree of the antenna and the module is poor or the quality of the antenna itself is problematic.

4.2 Module is easily damaged

- Please check the power supply to ensure that it is within the recommended power supply voltage. If it exceeds the maximum value, it will cause permanent damage to the module;
- Please check the stability of the power supply, and the voltage should not fluctuate significantly and frequently;
- Please ensure that the installation and use process is anti-static, and high-frequency components are electrostatically sensitive;
- Please ensure that the humidity should not be too high during installation and use, and some components are humidity sensitive devices;
- If there is no special requirement, it is not recommended to use at too high or too low temperature

4.3 Bit error rate is too high

- There is co-frequency signal interference nearby, stay away from the interference source or modify the frequency and channel to avoid interference;
- Unsatisfactory power supply may also cause garbled codes. Be sure to ensure the reliability of the power supply;
- Poor or too long extension cables and feeders can also cause high bit error rates.

Revise history

Version	Date	Revision description	Issued by
1.0	2021-09-22	Initial version	JH

About us

Technical support: support@cdebyte.com
Documents and RF Setting download link: www.ebyte.com
Thank you for using Ebyte products! Please contact us with any questions or suggestions: info@cdebyte.com
---
Phone: +86 028-61399028
Web: www.ebyte.com
Address: B5 Mould Park, 199# Xiqu Ave, High-tech District, Sichuan, China

Chengdu Ebyte Electronic Technology Co.,Ltd.