MBL Test Kit User manual

New generation package compatible Sub-1G wireless module

E49-900MBL-01



Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.PRODUCT OVERVIEW 2

1.1 PRODUCT INTRODUCTION 2
1.2 PIN DESCRIPTION 2
1.3 SUPPORT LIST 3

2.SOFTWARE INTRODUCTION 4

2.1 CATALOGUE STRUCTURE 4
2.2 IAR ENGINEERING 5
2.3 MAIN FUNCTIONS 6
2.4 SENDING AND RECEIVING TIMING 6
2.5 PROGRAMMING 8

3.QUICK DEMO 10

3.1 SIGNAL CABLE CONNECTIONS 10
3.2 SERIAL ASSISTANT 11

4.FAQ 13

4.1 UNSATISFACTORY TRANSMISSION DISTANCE 13
4.2MODULE VULNERABLE TO DAMAGE 13
4.3 TOO HIGH BER 13

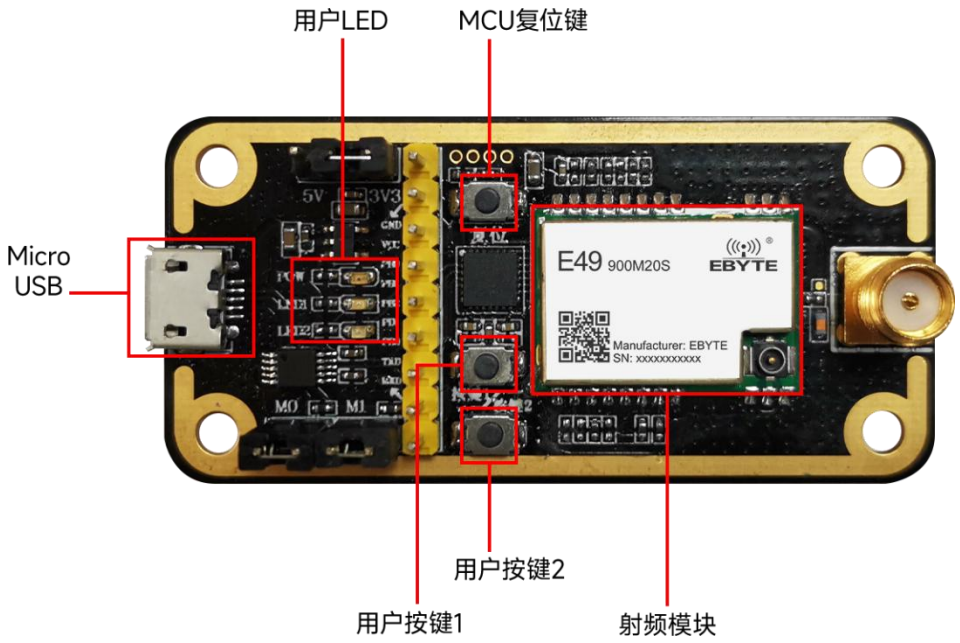
REVISE HISTORY 14

ABOUT US 14

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.Product Overview

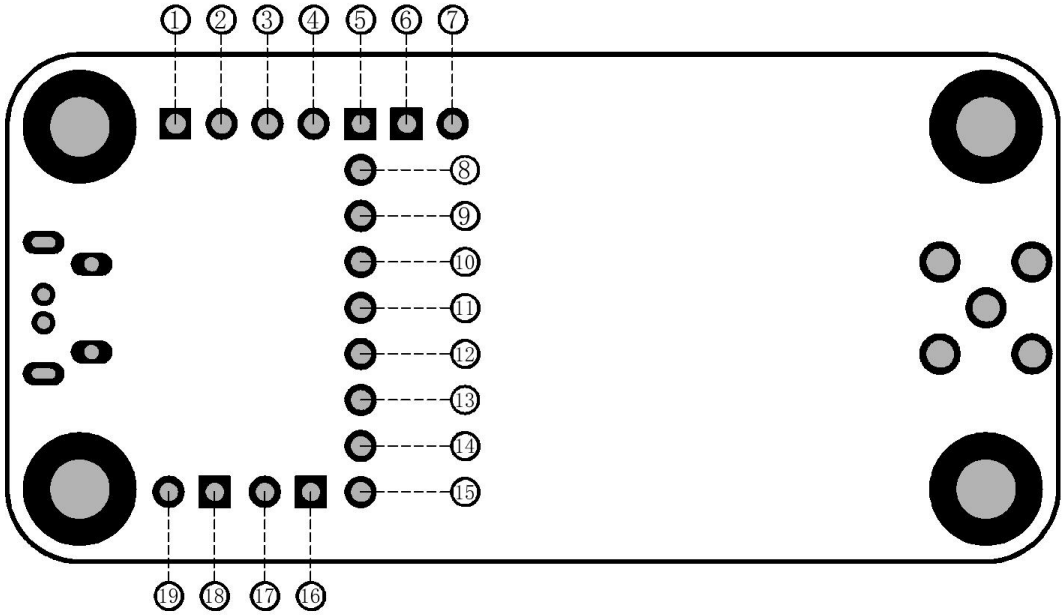


1.1 Product introduction

MBL series evaluation suite is designed to help users quickly evaluate Ebrand's new generation of packaged and compatible wireless modules. Most of the pins on the board have been drawn to both sides of the pins, developers can easily connect a variety of peripheral devices through jumpers according to actual needs.

This suite provides complete software application examples to help customers quickly start wireless data communication development. Different types of Sub-1G wireless modules can be onboard according to customer requirements. The supported modules all have pin-compatible packages for quick replacement.

1.2 Pin Description



Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Pin number	Name	Functions
1	VCC	The module power supply pin needs to be short connected to pin 2 to supply power to the module
2	3.3V	3.3V power pin
3	3.3V	3.3V power pin
4	VIO	MCU power supply pin, needs to be shortedto pin 3 to supply power to the MCU
5	GND	Base board reference ground
6	REST	MCU external reset pin
7	SWIM	MCU SWIM pin
8	VIO	MCU power supply pin
9	PC0	**Module reset pin**
10	PB7	Module MISO Pin
11	PB6	Module MOSI Pin
12	PB5	Module SCLK Pin
13	PB4	Module NSS Pin
14	TXD	MCU Serial TXD
15	RXD	MCU serial RXD
16	M1	Module mode switching pins (see module product manual for details)
17	GND	Base board reference ground
18	M0	Module mode switching pins (see module datasheet for details)
19	GND	Chassis reference ground

1.3 Support List

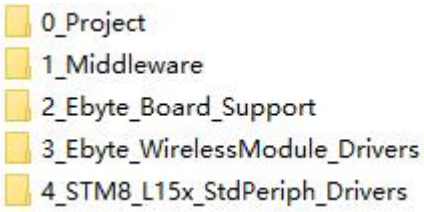
	RF IC	Manufacturer	Module model
1	CC1101	Texas Instruments	E07-400M10S
2	CC1101	Texas Instruments	E07-900M10S
3	SI4438	Silicon Labs	E30-400M20S
4	SI4463	Silicon Labs	E30-900M20S
5	LLCC68	Semtech	E220-400M22S
6	LLCC68	Semtech	E220-900M22S
7	SX1278	Semtech	E32-400M20S
8	SX1276	Semtech	E32-900M20S
9	SX1268	Semtech	E22-400M22S
10	SX1262	Semtech	E22-900M22S
11	AX5243	ON Semiconductor	E31-400M17S
12	LLCC68	Semtech	E220-400MM22S
13	LLCC68	Semtech	E220-900MM22S

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.Software Introduction

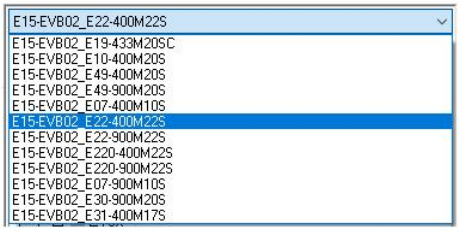
2.1 Catalogue structure

	事项	说明
1	File directory	A sample project can be downloaded from the website and opened in the directory as shown below:  0_Project 1_Middleware 2_Ebyte_Board_Support 3_Ebyte_WirelessModule_Drivers 4_STM8_L15x_StdPeriph_Drivers
2	Catalogue description	You can use the IAR For STM8 development environment to find the entry file and 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**

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2 IAR Engineering

	事项	说明
	Engineering structures	Using the IAR For STM8 development environment you can open the project and see the basic structure: 
	Switching workspaces	Global macro definitions and file paths are defined in the C/C++ Compiler options to distinguish between the driver files of different modules. When switching workspaces, different macro definitions will be used, thus switching the driver files of different modules:  Changed the Exclude from build attribute of Drivers/Ebyte/RF to select the target moduledriver folder for the compilation process. Changed Additional include in the project C/C++Compiler to specify the module driver file path. Changed Defined symbols in Project C/C++Compiler to define global macro definitions to help configure module driver properties: 

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

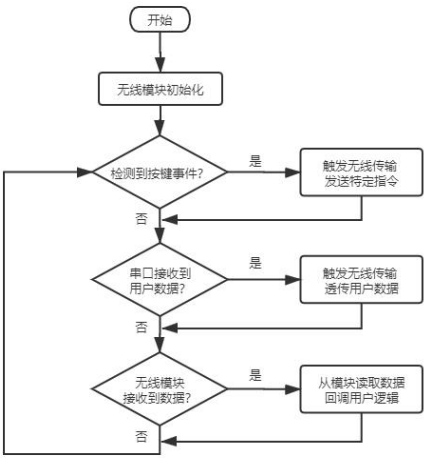
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3 Main functions

The main function entry is in main.c. The flow of the demonstration function is simplified as follows:

	Matters	Description
1	Key functions	If a key is pressed, the command data is sent wirelessly. In essence, this means sendinga specific string "ping" and expecting to receive a response "pong".
2	Serial data to wireless transmission	The serial port automatically starts transmitting data wirelessly when it receives the data, which of course contains some special command responses that are mainlyusedfor special tests and can be ignored by the user. Once the transmission is complete, theuser function is automatically called back so that the transmission logic can be handledby itself.
3	Receive data wirelessly	The underlying driver will copy the data and pass it to the user callback function, which will handle the receiving logic itself.

The software process is simplified as shown in the following diagram:



2.4 Sending and Receiving Timing

Wireless modules exist in a variety of operating states and can only perform a specific function in the correspondingstate. In terms of sending and receiving data in its simplest form, only the transmit mode and receive mode are considered.

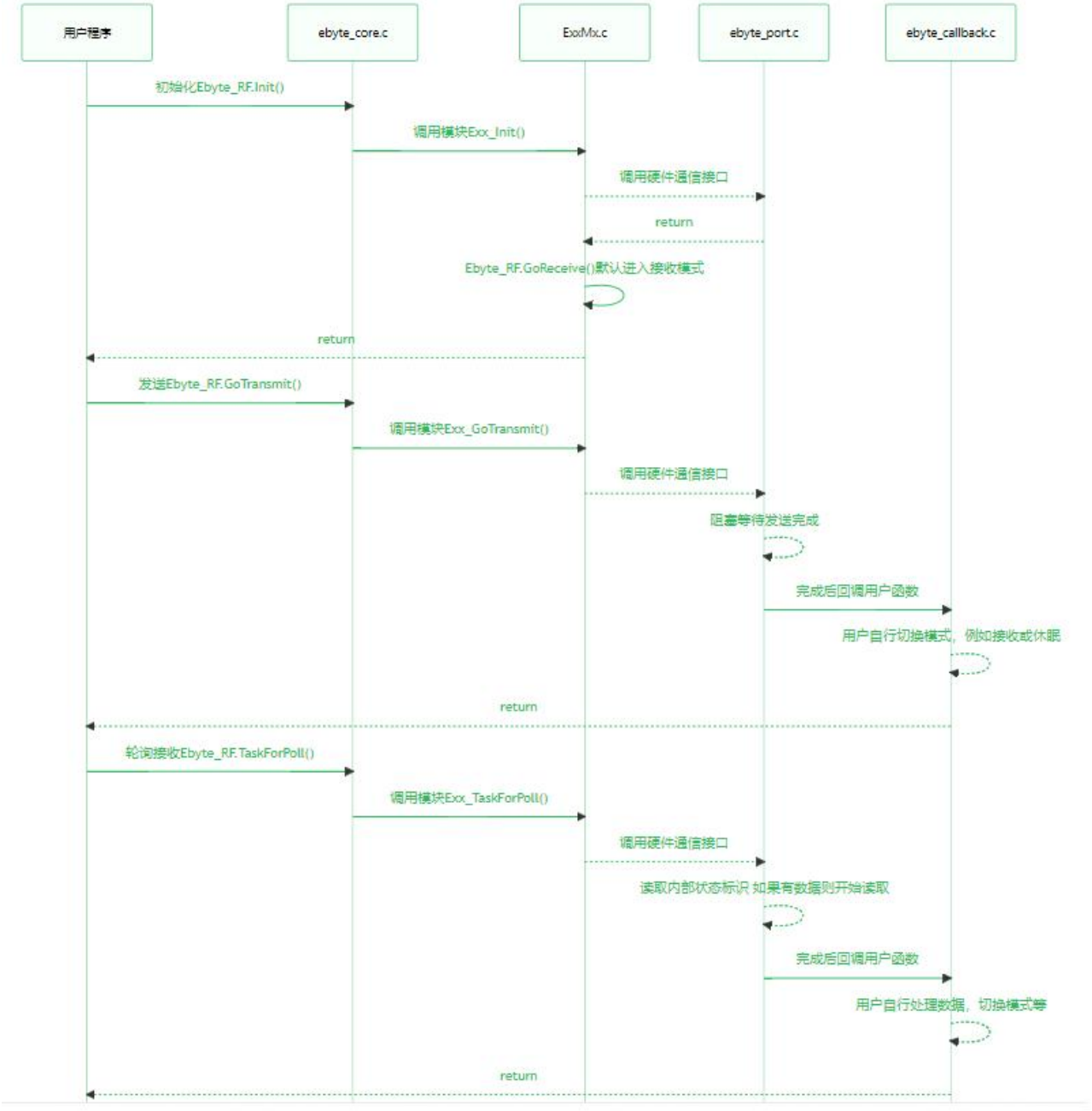
	Matters	Description
1	Receiving mode	Receive mode is automatically entered after the default initialisation. In essence, this means that the receive function is called during initialisation and thus enters receivemode. If you want to consider entering another mode after initialisation, such as sleep, you can simply replace it with the Go_xxxxx() function of the same type.
2	Sending mode	When the transmit function is called, the underlying driver actually switches the module into standby mode first, where the configuration of modulation

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

		parameters, such as frequency, power, frequency bias, etc., is usually done. After the parameters have been configured correctly, it gradually enters some intermediate modes, turningon the internal FIFO, PA, external XTAL, etc., and the current consumption graduallyclimbs. Finally it switches into transmit mode, triggering the wireless data transmission. After completion the module enters standby mode, this state cannot continue to send and receive and requires the user to handle the next mode in the callback function itself. When the function is complex and continuous receptionor continuous transmission is required, please further switch to other modes accordingtothe chip characteristics.

The timing diagram is shown below：



Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.5 Programming

	Doc.	Key description

1	ebyte_core.h	A module structure is defined, abstracting the basic functionality to which the functions of the underlying module will be bound. When used for simple sending andreceiving applications, it is not necessary to understand the underlying details of eachmodule's work and the abstracted functions can be called directly to start sending andreceiving data. If you need to customise some of the functions, you can also consider integrating them into the structure. If you know enough about the functions of the underlying modules, you can also just remove the ebyte_core.c/h file, there is no strongcoupling 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;`

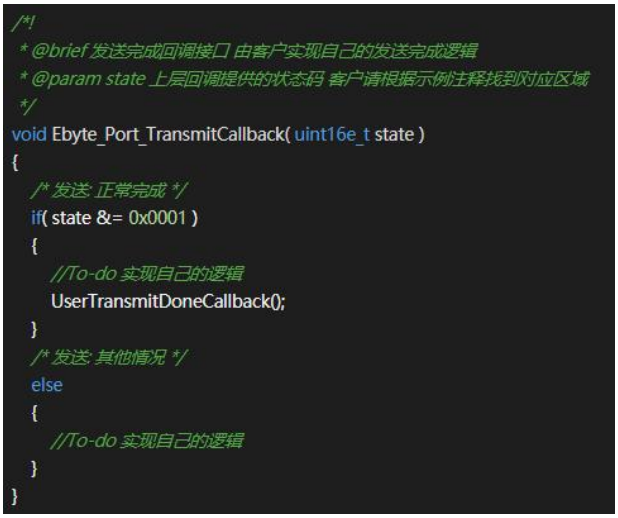
2	ebyte_exx.c	It is the specific module driver file, which is usually encapsulated and does not needto be changed by the user, only how the data is input and output from the "box".

3	ebyte_port.c	Specifically designed to bind SPI and GPIO on different hardware platforms, abstracted as inputs to a "box". The user needs to populate the communication interfaces in their hardware platform with fixed locations according to the comments. In general, thismeans providing the SPI send/receive functions and pin level control. Some modules areslightly special, for example the E49 uses half-duplex SPI, so if you are too lazy to writeacommunication driver then just bind the IO to a fixed position and leave the rest to themodule driver to emulate the IO itself to achieve communication. As shown in the diagram below, the SPI interface location is requested in the comments to fill in the specific send/receive function, the SPI send data is passed in by send and the SPI receivedata is returned by result.
		`/*!`
		` * @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 specifically designed to bind the user's own sending and receiving logic, abstracted as the output of a "box". Essentially, the module driver is calling the user's

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

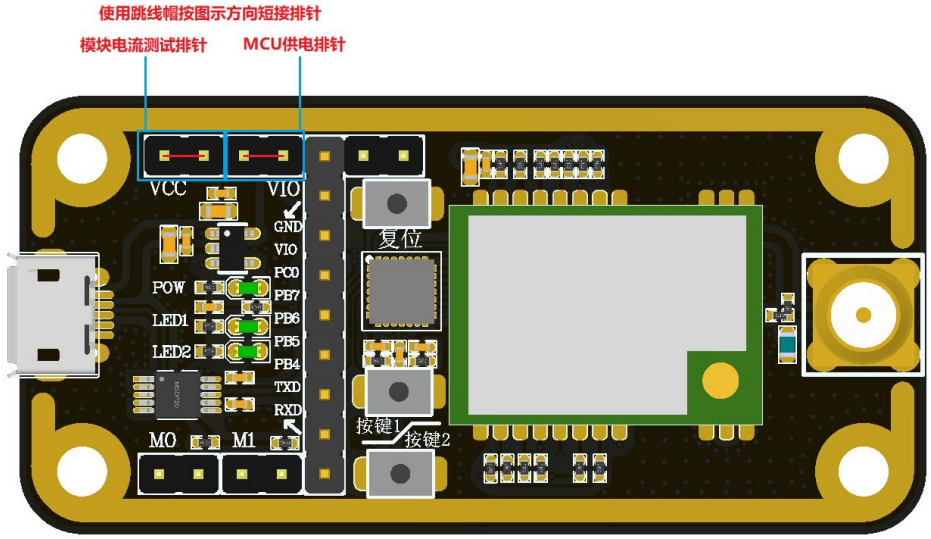
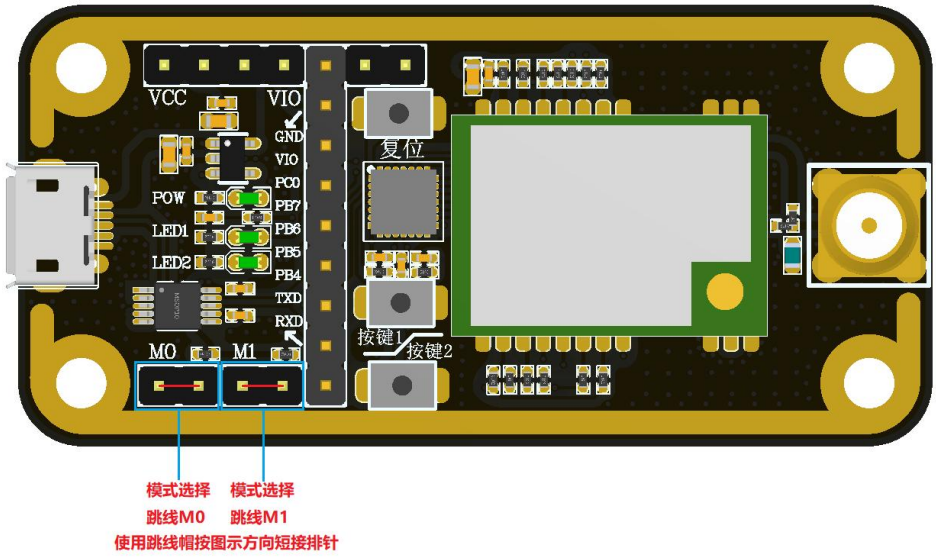
		callback function directly after it has determined that the sending or receiving is complete. As shown in the diagram below, the user's logic function is simply populated in the To-doprompt position. state is passed from the module driver and actually handled by the Exx_GoTransmit() function, which can be considered modified to support more cases when the functionality is complex. 
	ebyte_exx.h	Some general modulation parameters are defined, which generally do not needtobemodified and can be adjusted in them by yourself. Please note that the module driver hasarange check for the parameters and incorrect modulation parameters will lead to initialisation failure. The following example shows the FSK modulation parameters: 
	board.c	STM8 peripheral initialisation, involving SPI, TIMER, GPIO, etc., strongly coupled to the hardware used.
	board_button.c	The keystroke event queue, in terms of data structure, is a FIFO, and when the timer detects a keystroke it will store the corresponding event in the queue and wait for the mainloop to respond.
	board_mini_printf.c	Simplified printf, with reduced functionality but small footprint. The DEBUG macro in the project mainly relies on mprintf provided by this file.
	ebyte_kfifo.c	For serial port data reception, optimized universal FIFO queue, suitable for cache.
	ebyte_debug.c	It is used to connect to a PC for some tests. Generally, it is not needed.
	stm8l15x_it.c	All interrupt function entry, for serial port, timer, keystroke IO interrupt service functions are concentrated here.

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3.Quick Demo

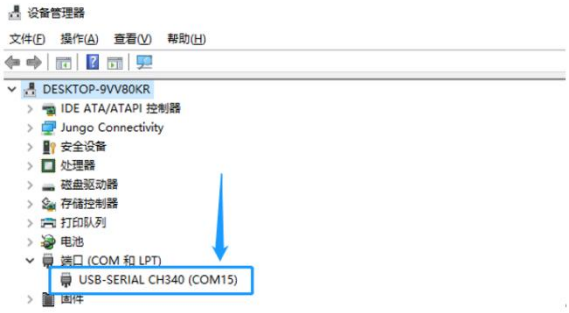
3.1 Signal cable connections

	Matters	Description
1	Power jumper caps	
2	Mode selection jumper cap	
3	Auxiliary	USB cable, antenna, PC, etc

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3.2 Serial Assistant

	Matters	Description
1	Device Manager View Serial Port Number	
2	Serial software	
3	Keystroke communication example	The #RECV identifier, used only for prompts, indicates data received by the wirelessmodule. The #SEND identifier, used only for hints, indicates data sent by the wireless module. 
4	Serial data transparent transmission	Serial data pass-through Direct transmission of the required content via XCOM

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.FAQ

4.1 Unsatisfactory transmission distance

- When linear communication barriers exist, the communication distance is attenuated accordingly.;
- Temperature, humidity, co-channel interference, which can lead to higher communication packet loss rates;
- The ground absorbs and reflects radio waves, making testing close to the ground less effective;
- The sea has a very strong ability to absorb radio waves, so the seaside testing effect is poor;
- The presence of metal objects near the antenna, or placed in a metal shell, signal attenuation will be very serious;
- Incorrect setting of the power register, too high setting of the air rate (the higher the air rate, the closer the distance);
- Low voltage of the power supply at room temperature is lower than the recommended value, the lower the voltagethe less power is generated.;
- Poor matching of the antenna to the module or the quality of the antenna itself.

4.2Module vulnerable to damage

- Please check the power supply to ensure that it is between the recommended supply voltages, as exceeding the maximum will result in permanent damage to the module;
- Please check the stability of the power supply, the voltage must not fluctuate significantly and frequently;
- Please ensure that the installation and use process is anti-static operation, high frequency devices are static sensitive;
- Please ensure that the humidity during installation and use is not too high, some components are humidity sensitive;
- If there is no special demand is not recommended to use at too high or too low temperature.

4.3 Too high BER

- Interference from nearby co-channel signals, stay away from sources of interference or modify the frequencyorchannel to avoid interference;
- Poor power supply may also cause garbled codes; be sure to ensure the reliability of the power supply;
- Poor quality or too long extension cables or feeders can also cause high BER.

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Revise History

Version	Revise date	Revision Notes	Maintained By
1.0	2021-09-22	Initial version	JH
1.1	2022-12-17	Added support for new models	JH

About us

Sales Hotline: 4000-330-990
Company Tel: 028- 61543675
Technical support: support@cdebyte.com
Official website: https://www.cdebyte.com
Company Address: Building B5, No. 199, West District Avenue, High-tech West District, Chengdu City, Sichuan Province

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

Copyright ©2012–2019，Chengdu Ebyte Electronic Technology Co.,Ltd

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000