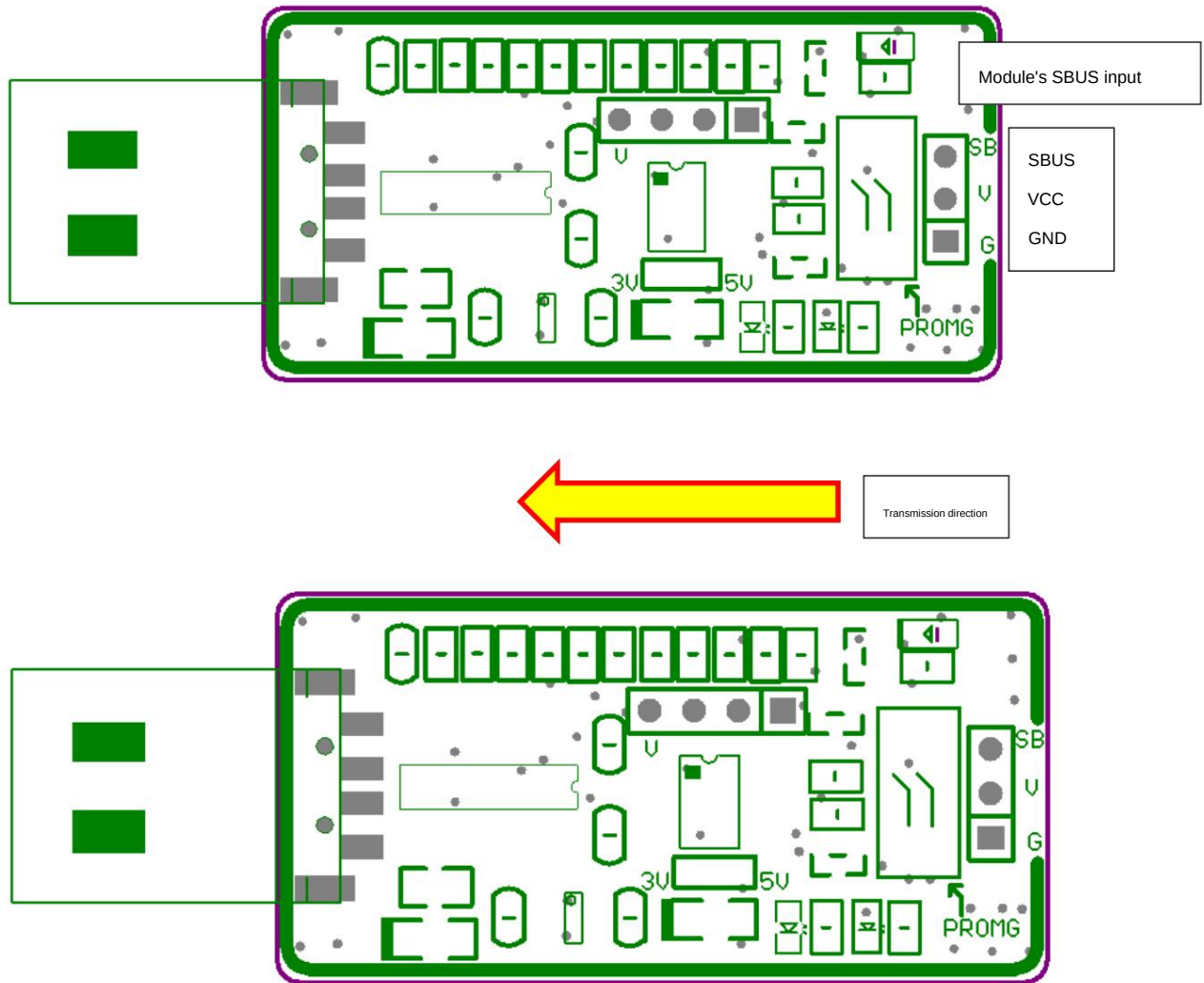
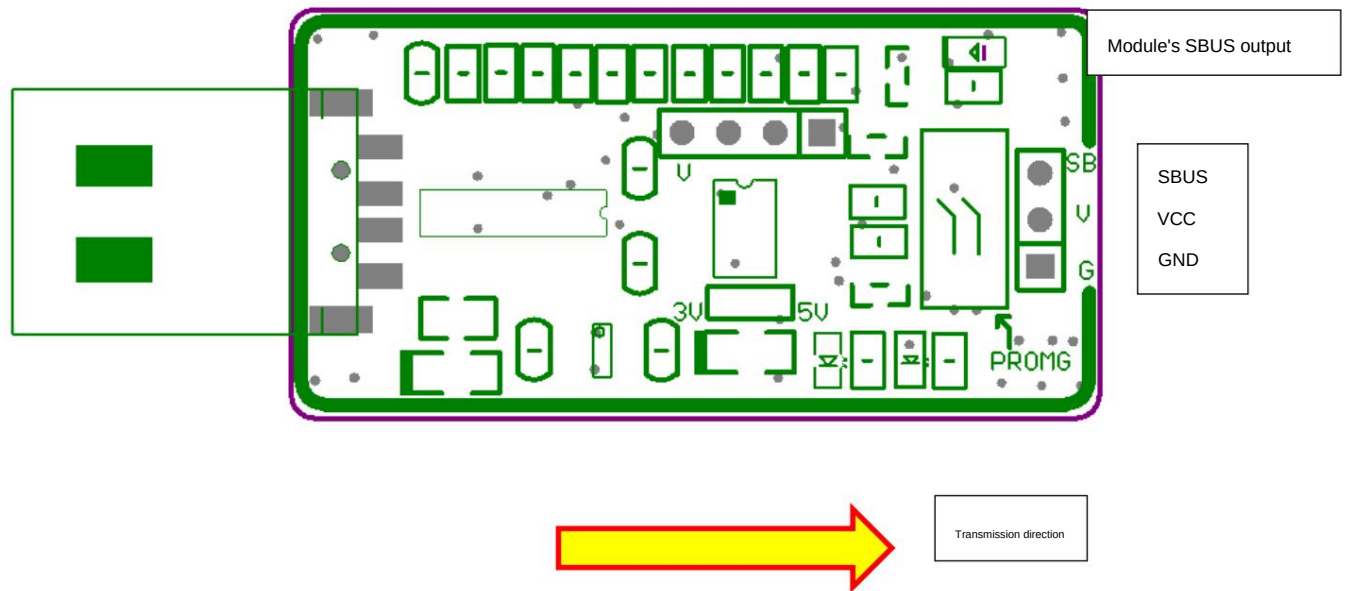


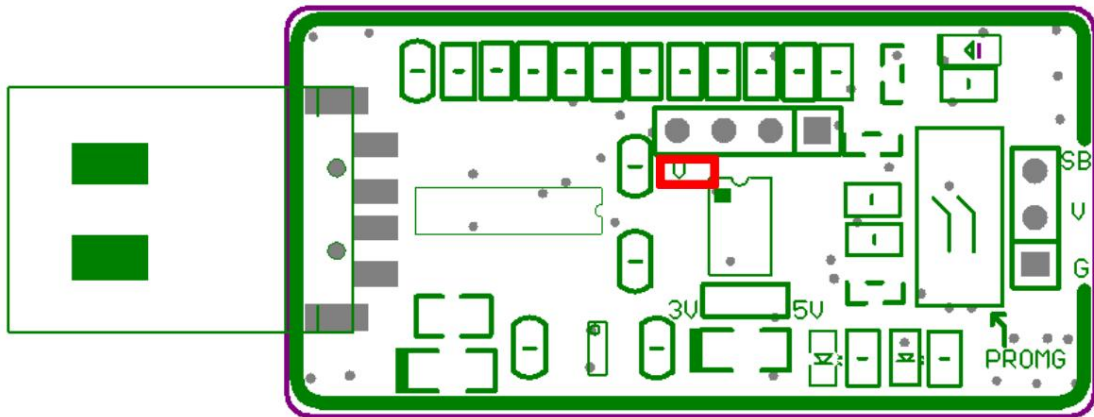
1. SBUS to USB Module: SBUS is the input terminal, and USB is the output terminal. The voltage can be selected via a jumper, with 3V and 5V options available.
- The module can supply 5V/50mA of power externally via VCC and GND.



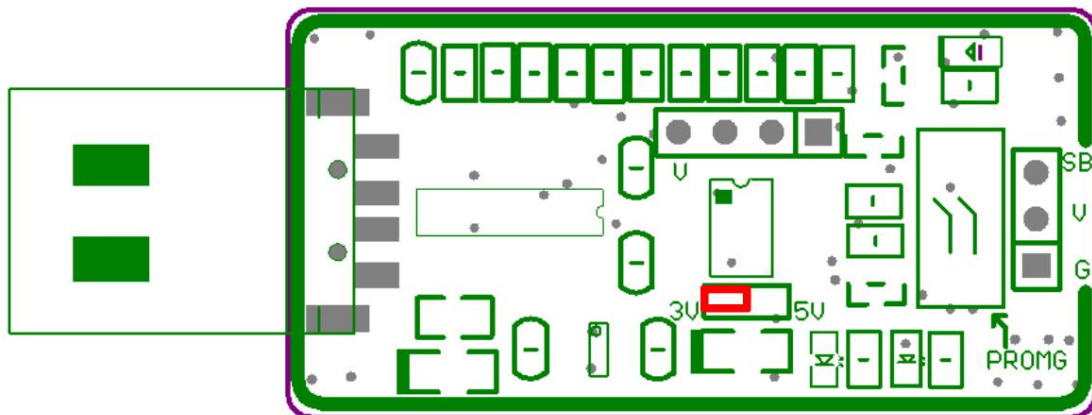
II. USB to SBUS Module: SBUS is the output terminal, and USB is the input terminal. The voltage can be selected via a jumper, with 3V and 5V options available. The module can supply 5V/50mA power externally via VCC and GND.



3. Power Supply Selection (Note that most receivers are 5V. If using USB power, the receiver cannot be connected to a separate power source. Our module can power the receiver. If both sides need to be powered simultaneously, the voltage on both sides must be identical; one side cannot be 5V and the other 3V.) The module's drive current is approximately 50mA. For higher current requirements, please consult customer service.



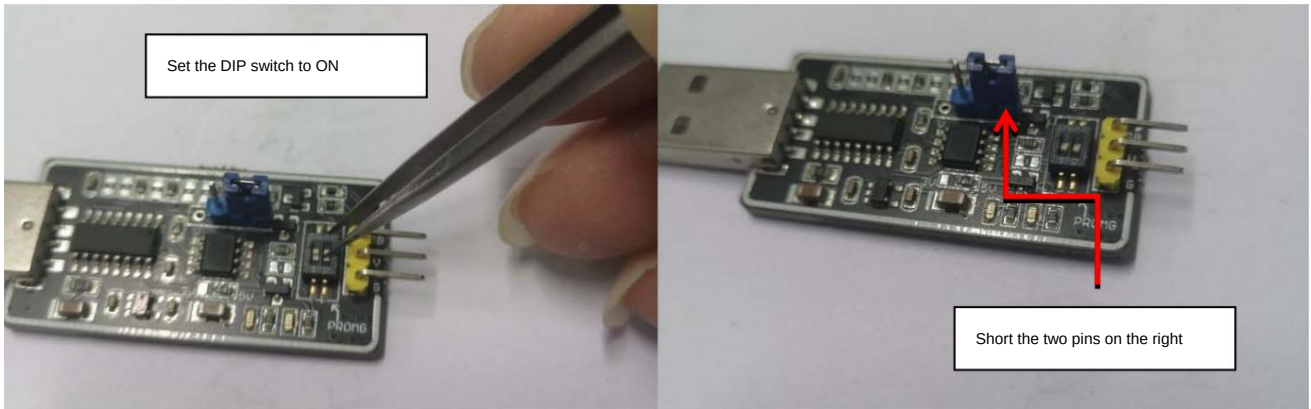
5V selection



3V selection

IV. Parameter Settings The

baud rate can be selected using a parameter tuning tool. Currently supported baud rates are 115200, 57600, 38400, and 9600. It consists of 1 start bit, 1 stop bit, 8 data bits, and no parity. For modules with adjustable frame rates, up to 14 frame rate settings from 10ms to 500ms can be configured. Step 1: Prepare a 3-pin header and a header cap (usually included with the module). If not, a pair of tweezers will suffice.



Since the pin header is not soldered on, it is best to press it down by hand to ensure that it can be shorted.



Press down on the pin header and insert it into the USB port. The board will then enter setup mode. After one second, you can remove the pin header. Now, directly open the serial port parameter tuning software and read the data from the current module. If you cannot read the data, repeatedly confirm the above steps.

Step 2: Return the DIP switch to its original position. (Make sure to remove the pin header first.)



Set both DIP switches to the Off position.

After the parameters are adjusted, be sure to switch both sets of DIP switches to the **OFF** position.



Select the corresponding COM number and click OPENCOM.





We can read the current baud rate and frame rate. If the frame rate is non-adjustable, it will display "non-adjustable"; if it's adjustable, the specific frame rate intervals will be displayed. We can also change the baud rate; currently, four baud rates are supported: 9600, 38400, 57600, and 115200. Fourteen frame rate increments from 10ms to 500ms are supported.

The indicator light will illuminate after each successful read or write operation; otherwise, communication will fail. In this case, check if the chip has truly entered setup mode and repeat the steps above until the correct data is read.

After the parameters are adjusted, be sure to switch both sets of DIP switches to the **OFF** position.

VI. Debugging Assistant

Launch our debugging assistant (Sbus Display Console.exe, select Sbus Display Console with Decoding.exe for modules with decoding).



SBUS includes 16 channels of data, with values ranging from 0 to 2048. The defined range may vary depending on the remote control, but the basic data format remains the same.

We can drag the slider to change the data output of each channel.

7. Software Agreement

- 1. An SBUS without decoding is 25 bytes long, representing 16 channels. The format is as follows:

Start Byte = 0x0F. The middle 22 bytes contain data for 16 channels, each channel represented by 11 bits, ranging from 0 to 2047.

The End Byte varies depending on the S-BUS protocol version.

Definition of Flags: bit7

= ch17 = digital channel (0x80) bit6 = ch18 = digital channel

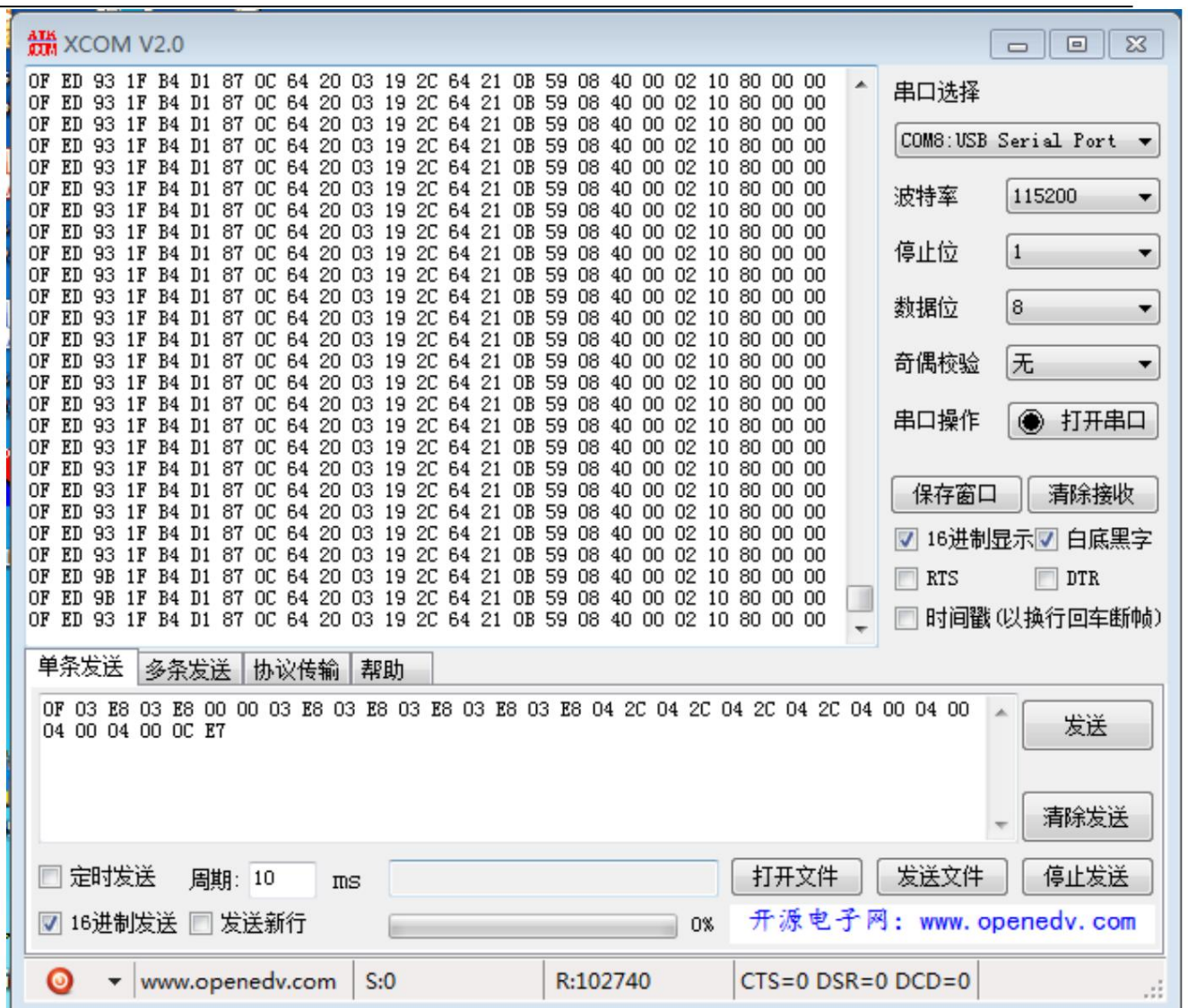
(0x40) bit5 = Frame lost, equivalent red LED on receiver (0x20)

bit4 = failsafe activated (0x10) bit3 = n/a bit2 = n/a bit1 = n/a bit0 = n/a

The data is as follows

0F ED 93 1F B4 D1 87 0C 64 20 03 19 2C 64 21 0B 59 08 40 00 02 10 80 00 00 (Total 25 bytes, no checksum)

Section, compressed data format.



2: Output protocol with encoding/decoding module:

The module with decoding capability has completed the **SBUS** protocol parsing. The output is 35 bytes, in the following format.

Frame header + 16 channels (32 bytes total) + flag + XOR

Start Byte = 0x0F. The middle 32 bytes contain data for 16 channels, each channel using an integer ranging from 0 to 2047. The high byte comes first, followed by the low byte.

XOR (checksum) is a checksum performed by performing an XOR operation on all 33 bytes except the header byte.

Definition of Flags:

bit7 = ch17 = digital channel (0x80) bit6 = ch18 = digital

channel (0x40) bit5 = Frame lost, equivalent red LED on

receiver (0x20) bit4 = failsafe activated (0x10) bit3 = n/a bit2 = n/a bit1 = n/a

bit0 = n/a

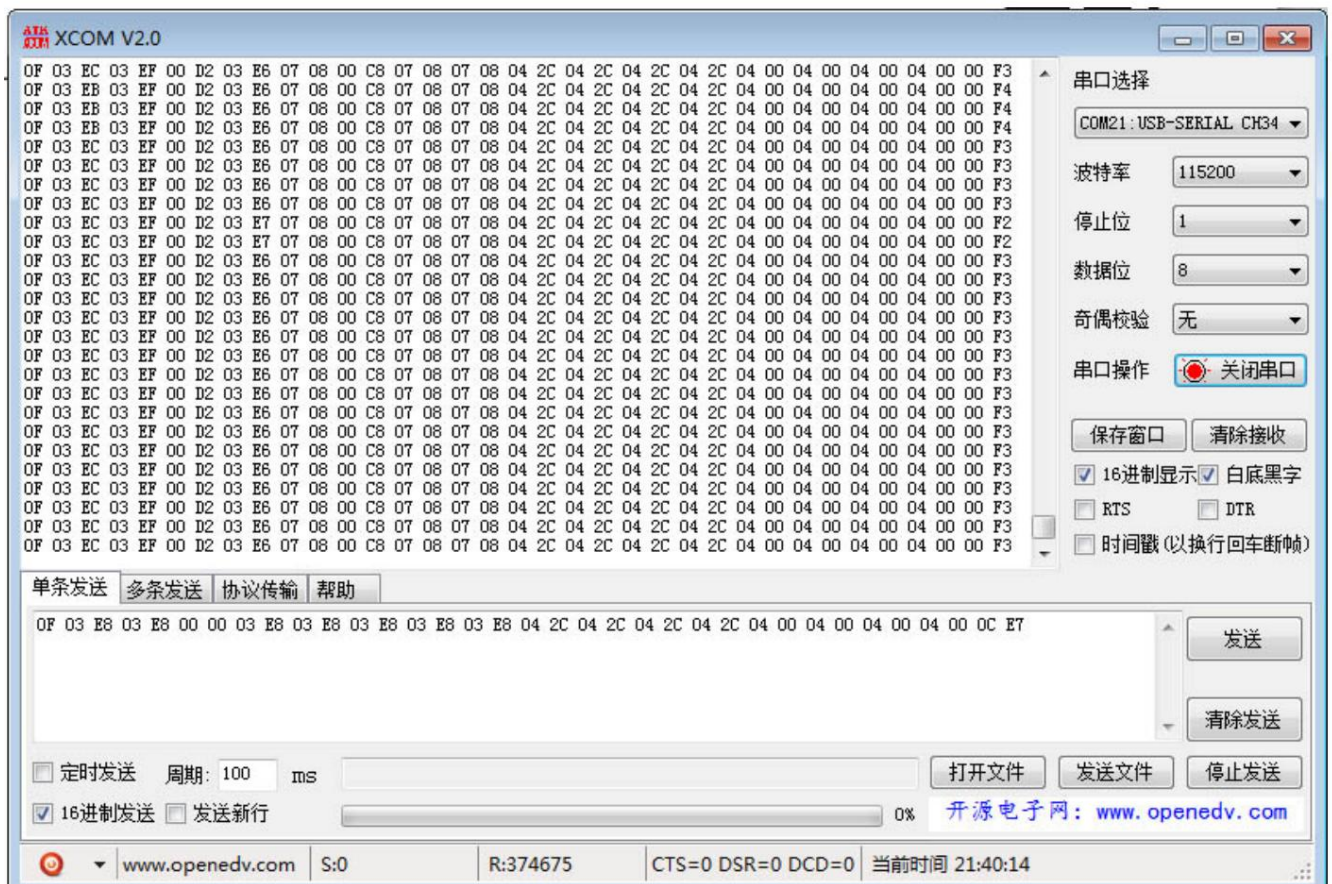
The following is a data frame

0F 03 E8 03 E8 00 00 03 E8 03 E8 03 E8 03 E8 04 2C 04 2C 04 2C 04 2C 04 00 04 00 04 00 04
00 0C E7

0F is the frame header, E7 is the checksum, and 0C is the flag.

First channel value: 03E8 (hexadecimal) = 1000

Sixteenth channel value: 0400 = 1024



8: Serial Port Programming

Serial port interrupts can be used. Three main parameters need to be controlled.

1. The first byte of any received frame should always be 0F. Otherwise, discard it.
2. After receiving 35 bytes, a verification process must be performed. If the verification passes, the data is considered valid.
- 3: Timeout Detection. The interval between frames is known to you and is fixed. If a frame times out before 35 characters have been received, the timeout will occur.

If the entire frame is discarded, the entire frame is discarded. The index of the receive array is then reset.