



EC-JET Communication Protocol

Version v3.3

Contents

Contents.....	2
Version supported.....	4
Instructions supported by this version of the protocol	4
Printers supported by this version of the protocol	6
Version update record	7
General frame format.....	8
Frame structure.....	8
Byte escape	8
Check word calculation.....	9
Command send and response mode	12
Detailed instructions.....	13
Set Print Width.....	13
Get Print Width.....	13
Set Print Delay.....	13
Get Print Delay.....	14
Set Print Interval.....	14
Get Print Interval.....	14
Set Print Height.....	15
Get Print Height.....	15
Set Print Count.....	16
Get Print Count.....	16
Set Reverse Message.....	17
Get Reverse Message.....	17
Set Trigger Repeat.....	18
Get Trigger Repeat.....	18
Get Printer Status.....	19
Set Printer Head Code.....	19
Get Printer Head Code.....	20
Set Photocell Mode.....	20
Get Photocell Mode.....	21
Get Jet Status.....	22
Get System Times.....	23
Start Jet.....	24
Stop Jet.....	24
Start Print.....	24
Stop Print.....	25
Trigger Print.....	25
Set Date Time.....	26
Get Date Time.....	26
Get Font List.....	27
Get Message List.....	28
Create Field (Text).....	29
Create Field (Barcode).....	30

Create Field (Logo).....	31
Create Field (Remote Text).....	32
Create Field (Remote Barcode).....	33
Create Field (DateTime Text).....	34
Create Field (DateTime Barcode).....	36
Create Field (SerialNum Text).....	37
Create Field (SerialNum Barcode).....	39
Download Remote Buffer.....	40
Delete Last Field.....	40
Delete Message Content.....	41
Set Current Message.....	41
Set Aux Mode.....	42
Get Aux Mode.....	42
Set Reference Modulation.....	42
Get Reference Modulation.....	43
Reset Serial Number.....	43
Reset Count Length.....	43
Print Trigger State.....	44
Print Go State.....	44
Print End State.....	44
Request Remote Data.....	45
Print Fault State.....	45
Communication line.....	46
RS232Communication.....	46

Version Supported

Instructions supported by this version of the protocol

- Set Print Width ID: 0001h
- Get Print Width ID: 0002h
- Set Print Delay ID: 0003h
- Get Print Delay ID: 0004h
- Set Print Interval ID: 0005h
- Get Print Interval ID: 0006h
- Set Print Height ID: 0007h
- Get Print Height ID: 0008h
- Set Print Count ID: 0009h
- Get Print Count ID: 000Ah
- Set Reverse Message ID: 000Bh
- Get Reverse Message ID: 000Ch
- Set Trigger Repeat ID: 000Dh
- Get Trigger Repeat ID: 000Eh
- Get Printer Status ID: 000Fh
- Set Print Head Code ID: 0010h
- Get Print Head Code ID: 0011h
- Set Photocell Mode ID: 0012h
- Get Photocell Mode ID: 0013h

- Get Jet Status ID: 0014h
- Get System Times ID: 0015h
- Start Jet ID: 0016h
- Stop Jet ID: 0017h
- Start Print ID: 0018h
- Stop Print ID: 0019h
- Trigger Print ID: 001Ah
- Set Date Time ID: 001Bh
- Get Date Time ID: 001Ch
- Get Font List ID: 001Dh
- Get Message List ID: 001Eh
- Create Field (Text) ID: 001Fh
- Create Field (Barcode) ID: 001Fh
- Create Field (Logo) ID: 001Fh
- Create Field (Remote Text) ID: 001Fh
- Create Field (Remote Barcode) ID: 001Fh
- Create Field (DateTime Text) ID: 001Fh
- Create Field (DateTime Barcode) ID: 001Fh
- Create Field (SerialNum Text) ID: 001Fh
- Create Field (SerialNum Barcode) ID: 001Fh
- Download Remote Buffer ID: 0020h
- Delete Last Field ID: 0021h

- Delete Message Content ID: 0022h
- Set Current Message ID: 0023h
- Set AUX MODE ID: 0024H
- Get AUX Mode ID: 0025h
- Set Shaft Encoder Mode ID: 0026h
- Get Shaft Encoder Mode ID: 0027h
- Set Reference Modulation ID: 0028h
- Get Reference Modulation ID: 0029h
- Reset Serial Number ID: 002Ah
- Reset Count Length ID: 002Bh
- * Print Trigger State¹ ID: 1000h
- * Print Go State ID: 1001h
- * Print End State ID: 1002h
- * Request Remote Data ID: 1003h
- * Print Fault State ID: 1004h

Printers supported by this version of the protocol

Printer Model: EC-2000

Version of software: from 180326

¹. The message with the * is the message sent by the printer to the PC.

Version Update Record

Version No.	date	updated contents	Applicable printer software version
V3.0	2017.06.30	original version	From 170630
V3.1	2017.07.26	Adding printer send commands to the host	From 170726
V3.2	2018.01.23	Adding set current data command and adding send a command to the host when printer failure occurs.	From 180123
V3.3	2018.03.26	Adding the reset sequence number and the reset meter counting segment command. Fixing two command number errors.	From 180326

General frame format

Frame structure

All communication is in units of frames, and the frame structure is as follows² :

[STX][ADDR][CMD-ID][DAT-OFFSET][CMD-INF][<DATA>][<CHECKSUM>][ETX]

[STX] Fixed data 7Eh, indicating the beginning of the frame, 1 byte.

[ADDR] Multi-machine communication address of the printer, 1 byte.

[CMD-ID] Instruction ID, indicating the type of instruction, 2 bytes³.

[DAT-OFFSET] is fixed at 00h 0Ch, 2 bytes.

[CMD-INF] Command additional information, 7 bytes.

When the PC sent to the printer, the 7 bytes are meaningless and fixed at 00h.

When the printer sent to the PC, the data structure of this field is as follows:

[ACK] 1 byte, data frame reception status, 06h for reception completion, and 15h for frame error.

[NR] 2 bytes.

[DEV_STATUS] 2 bytes.

[CMD_STATUS] 2 bytes,

0 means the command was successfully executed.

1 means the command execution failed.

2 means that the command number is not implemented in the software version.

4 means the jet is not running

8 means a parameter error

10 means the printer is busy

[<DATA>] Instruction accompanying data. The length is not fixed.

[<CHECKSUM>] Checksum, length 0, 1 or 2 bytes.

[ETX] Fixed data 7Fh, indicating the end of the frame, 1 byte.

Byte escaping

In the frame structure, 7Eh, 7Fh can only be used in the [STX][ETX] bit, they must not appear in other positions.

The specified escaping flag is 7Dh, except for [STX][ETX]:

Single-byte data 7Eh is escaped to double-byte 7Dh, 5Eh.

Single-byte data 7Dh is escaped to double-byte 7Dh, 5Dh.

Single-byte data 7Fh is escaped to double-byte 7Dh, 5Fh.

² [[XX] indicates fixed length data, and [<XX>] indicates indefinite length data.

³ [When multi-byte data is transmitted, the low bit is sent first, and then the high bit is transmitted.].

Check word calculation

Printer Menu Settings - Remote Communication - Verification mode, you can set the communication frame check word mode.

none

Do not use a check word.

The length of the <<CHECKSUM>> in the communication frame is 0 bytes.

Mod256

Using Mod256 as the check method, the checksum is calculated by summing all the bytes starting from [ADDR] to [<DATA>] in the communication frame, and modulo the 256, which is the check word.
[<CHECKSUM>] is 1 byte in length.

rc16

Using CRC16 as the check method, the input of CRC16 is all bytes in the communication frame starting from [ADDR] to [<DATA>], and the output is the check word.
[<CHECKSUM>] is 2 bytes in length.

The CRC16 algorithm used in this version of protocol is: CRC-16/X25.

Polynomial: . (0x1021).

Initial value: 0xFFFF.

Data reversal: LSB First.

XOR value: 0xFFFF.

CRC16 direct calculation code:

```
/*****************************************************************************  
* Name:  CRC-16/X25      x16+x12+x5+1  
* Poly:  0x1021  
* Init:  0xFFFF  
* Refin: True  
* Refout: True  
* Xorout: 0xFFFF  
*****/  
  
uint16_t crc16_x25(uint8_t *data, uint16_t length) {  
    uint16_t crc = 0xffff; // Initial value  
    while(length--) {  
        crc ^= *data;  
        data++;  
        for (uint8_t i = 0; i < 8; ++i) {  
            if (crc & 1) {  
                crc = (crc >> 1) ^ 0x8408; // 0x8408 = reverse 0x1021  
            } else {  
                crc = (crc >> 1);  
            }  
        }  
    }  
    return ~crc; // crc ^ Xorout  
}
```

CRC16 lookup table calculation code:

```
const uint16_t crcatab16[] = {  
    0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,  
    0x8e48, 0x9dc1, 0xaf5a, 0xbcd3, 0xca6c, 0dbe5, 0xe97e, 0xf8f7,  
    0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,  
    0x9cc9, 0x8d40, 0xbfd8, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,  
    0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,  
    0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,  
    0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,  
    0xbdc8, 0xae42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,  
    0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,  
    0xce4c, 0xdfc5, 0xed5e, 0xfcfd, 0x8868, 0x99e1, 0xab7a, 0xbaf3,  
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,  
    0xdeed, 0xcf44, 0xfddc, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,  
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,  
    0xef4e, 0xfc7, 0xcc5c, 0xdd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,  
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,  
    0xffcf, 0xee46, 0xdedd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,  
    0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,  
    0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,  
    0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,  
    0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,  
    0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,  
    0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,  
    0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,  
    0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,  
    0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
```

```
0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78,
};

// Calculates the 16-bit CRC for a given length of data.
uint16_t GetCrc16(const char *pData, uint32_t nLength){
    uint16_t fcs = 0xffff; // initialize
    while (nLength > 0) {
        fcs = (fcs >> 8) ^ crctab16[(fcs ^ *pData) & 0xff];
        nLength--;
        pData++;
    }
    return ~fcs; // negate
}
```

Command Sending and Response Mode

All communication is in units of frames. The master encapsulates the instructions and accompanying data into a frame and sends them to the printer. After the printer executes the instructions, the instructions and the accompanying data are also encapsulated into a frame and sent to the host for response. A complete instruction transmission and response process is as follows:

PC->Printer: 7E 00 16 00 0C 00 00 00 00 00 00 00 C3 A4 7F

Printer->PC: 7E 00 16 00 0C 00 06 00 00 00 00 00 00 0E FC 7F

After receiving the response frame, the master can judge whether the instruction is successfully executed according to [ACK], and can extract instructions and accompanying parameter from [<DATA>].

Detailed Instructions

Set Print Width

CMD-ID	0001H
Description	Set print width
2 bytes print width value	2 bytes print width value
Return [<DATA>] format	empty

Get Print Width

CMD-ID	0002H
Description	Get print width
Send [<DATA>] format	empty
Return [<DATA>] format	2 bytes print width value

Set Print Delay

CMD-ID	0003H
Description	Set print delay
Send [<DATA>] format	2 bytes print delay value
Return [<DATA>] format	empty

Get Print Delay

CMD-ID	0004H
Description	Get print delay
Send [<DATA>] format	empty
Return [<DATA>] format	2 bytes print delay value

Set Print Interval

CMD-ID	0005H
Description	Set the print interval
Send [<DATA>] format	2 bytes print interval value
Return [<DATA>] format	empty

Get Print Interval

CMD-ID	0006H
Description	Get the print interval
Send [<DATA>] format	empty
Return [<DATA>] format	2 bytes print delay value

Set Print Height

CMD-ID	0007H
Description	Set the print height
Send [<DATA>] format	1 bytes print height value (110 – 230)
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 07 00 0C 00 00 00 00 00 00 00 00 96 79 65 7F]

Printer -> PC: [7E 00 07 00 0C 00 06 00 00 00 00 00 00 DA D8 7F]

Get Print Height

CMD-ID	0008H
Description	Get the print height
Send [<DATA>] format	empty
Return [<DATA>] format	1 bytes print delay value (110 – 230)

Example:

PC -> Printer: [7E 00 08 00 0C 00 00 00 00 00 00 00 00 5B 9C 7F]

Printer -> PC: [7E 00 08 00 0C 00 06 00 00 00 00 00 00 96 BC F0 7F]

Set Print Count

CMD-ID	0009H
Description	Set print count
Send [<DATA>] format	<p>The 5-byte data structure is as follows:</p> <p>[CountType] 1 byte count type</p> <p>0 represents the total print count of the printhead</p> <p>1 represents the printing data printing count</p> <p>2 represents editing data printing count</p> <p>[PrintCount] 4 bytes print count value</p>
Return [<DATA>] format	empty

Example:

02 0C 00 00 00 AE 8B 7F]

Printer -> PC: [7E 00 09 00 0C 00 06 00 00 00 00 00 00 00 07 91 7F]

Get Print Count

CMD-ID	000AH
Description	Get print count
Send [<DATA>] format	<p>1 byte count type</p> <p>0 represents the total print count of the nozzle</p> <p>1 represents the printing data printing count</p> <p>2 represents editing data printing count</p>
Return [<DATA>] format	4 bytes print count value

Example:

PC -> Printer:[7E 00 0A 00 0C 00 00 00 00 00 00 00 00 02 1B 3D 7F]

Printer -> PC: [7E 00 0A 00 0C 00 06 00 00 00 00 00 00 A2 01 00 00 B3 61 7F]

Set Reverse Message

CMD-ID	000BH
Description	Set reverse message printing mode
Send [<DATA>] format	The 2-byte data structure is as follows: [Vertical Revert] 1byte [Horizontal Revert] 1byte
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 0B 00 0C 00 00 00 00 00 00 00 01 01 5B 60 7F]

Printer -> PC: [7E 00 0C 00 0C 00 06 00 00 00 00 00 00 25 3A 7F]

Get Reverse Message

CMD-ID	000CH
Description	Get reverse message printing mode
Send [<DATA>] format	empty
Return [<DATA>] format	The 2-byte data structure is as follows: [Vertical Revert] 1byte [Horizontal Revert] 1byte

Example:

PC -> Printer: [7E 00 0C 00 0C 00 00 00 00 00 00 00 00 0E C2 7F]

Printer -> PC: [7E 00 0C 00 0C 00 06 00 00 00 00 00 00 00 01 DF C5 7F]

Set Trigger Repeat

CMD-ID	000DH
Description	Set trigger repeat
Send [<DATA>] format	1 byte Trigger repetition number (minimum 1)
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 0D 00 0C 00 00 00 00 00 00 00 00 01 18 8D 7F]

Printer -> PC: [7E 00 0D 00 0C 00 06 00 00 00 00 00 00 52 CF 7F]

Get Trigger Repeat

CMD-ID	000EH
Description	Get trigger repeat
Send [<DATA>] format	empty
Return [<DATA>] format	1 byte Trigger repetition number (minimum 1)

Example:

PC -> Printer: [7E 00 0E 00 0C 00 00 00 00 00 00 00 00 2C 69 7F]

Printer -> PC: [7E 00 0E 00 0C 00 06 00 00 00 00 00 00 01 47 17 7F]

Get Printer Status

CMD-ID	000FH
Description	Get printer status
Send [<DATA>] format	empty
Return [<DATA>] format	<p>The 5-byte data structure is as follows:</p> <p>[Working Status] 1 byte</p> <p>1 jet stop 2 jet start 4 printing</p> <p>[Warning Status] 4 bytes Warning message</p> <p>4 bytes total 32 bits, from 0 to 31 bits for warning messages 3.00 to 3.31 respectively</p> <p>For example, 0001h means there is a warning message 3.00</p> <p>0002h indicates that there is a warning message 3.01</p> <p>0003h indicates that there are warning messages 3.00 and 3.01</p>

Example:

PC -> Printer: [7E 00 0F 00 0C 00 00 00 00 00 00 00 00 00 00 BD 3C 7F]

01 00 00 00 00 C8 3A 7F]

Set Printer Head Code

CMD-ID	0010H
Description	Set print head code
Send [<DATA>] format	14 bytes print head code ASCII value For example, the print head code is "12108010001701" The data is 31h 32h 31h 30h 38h 30h 31h 30h

	30h 30h 31h 37h 30h 31h
Return [<DATA>] format	empty

Example:

PC -> Printer: 7E 00 10 00 0C 00 00 00 00 00 00 00 00 00 00 00

31 32 31 30 38 30 31 30 30 30 31 37 31 32 05 03 7F]

Printer -> PC: [7E 00 10 00 0C 00 06 00 00 00 00 00 00 00 79 09 7F]

Get Printer Head Code

CMD-ID	0011H
Description	Get print head code
Send [<DATA>] format	empty
Return [<DATA>] format	<p>14 bytes print head code ASCII value For example, the print head code is "12108010001701" The data is 31h 32h 31h 30h 38h 30h 31h 30h 30h 30h 31h 37h 30h 31h</p>

Example:

PC -> Printer: [7E 00 11 00 0C 00 00 00 00 00 00 00 00 00 25 04 7F]

Printer -> PC: 7E 00 11 00 0C 00 06 00 00 00 00 00 00 00 00 00

31 32 31 30 38 30 31 30 30 30 31 37 30 31 55 9F 7F]

Set Photocell Mode

CMD-ID	0012H
Description	Set Photocell mode
Send [<DATA>] format	 1byte 0 interior trigger 1 photocell edge trigger 2 photocell level trigger 3 remote
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 12 00 0C 00 00 00 00 00 00 00 00 00 03 A6 33 7F]

Printer -> PC: [7E 00 12 00 0C 00 06 00 00 00 00 00 00 00 5B A2 7F]

Get Photocell Mode

CMD-ID	0013H
Description	Get Photocell mode
Send [<DATA>] format	empty
Return [<DATA>] format	1byte 0 interior trigger 1 photocell edge trigger 2 photocell level trigger 3 remote

Example:

PC -> Printer: [7E 00 13 00 0C 00 00 00 00 00 00 00 00 00 07 AF 7F]

Printer -> PC: [7E 00 13 00 0C 00 06 00 00 00 00 00 00 00 03 42 AB 7F]

Get Jet Status

CMD-ID	0014H
Description	Get jet status
Send [<DATA>] format	empty
Return [<DATA>] format	<p>The 10 bytes of data structure are as follows:</p> <p>[RefPress] 1 byte Reference pressure [Press] 1 byte Set pressure [ReadPress] 1 byte read pressure [SolventAddtion] 1 byte Solvent addition pressure [Modulation] 1 byte modulation [Phase] 1 byte Phase [RefVOD] 2 bytes reference ink speed [VOD] 2 bytes ink speed</p>

Example:

PC -> Printer:[**7E 00 14 00 0C 00 00 00 00 00 00 E1 0F 7F**]

Printer -> PC: [**7E 00 14 00 0C 00 06 00 00 00 00 00 00 00 00**

AA AA 00 AE 83 0C 59 52 00 00 E9 09 7F]

Get System Times

CMD-ID	0015H
Description	Get system times
Send [<DATA>] format	empty
Return [<DATA>] format	<p>32 bytes of data structure are as follows</p> <p>[PowerOnHour] 4 bytes Boot hours</p> <p>[PowerOnMinute] 4 bytes Power on minutes</p> <p>[JetRunningHour] 4 bytes Jet running hours</p> <p>[JetRunningMinute] 4 bytes Jet running minutes</p> <p>[FilterChangeHour] 4 bytes Main filter replacement remaining hours</p> <p>[FilterChangeMinute] 4 bytes Main filter replacement remaining minutes</p> <p>[ServiceHour] 4 bytes Service time remaining hours</p> <p>[ServiceMinue] 4 bytes Service time remaining minutes</p>

Example:

PC -> Printer: [7E 00 15 00 0C 00 00 00 00 00 00 00 00 00 00 70 5A 7F]

1B 00 00 00 03 00 00 00

0D 00 00 00 30 00 00 00

92 0F 00 00 0C 00 00 00

92 0F 00 00 0C 00 00 00

74 A0 7F 1

Start Jet

CMD-ID	0016H
Description	Start jet
Send [<DATA>] format	Empty
Return [<DATA>] format	Empty

Example:

PC -> Printer: [7E 00 16 00 0C 00 00 00 00 00 00 00 00 00 C3 A4 7F]

Printer -> PC: [7E 00 16 00 0C 00 06 00 00 00 00 00 00 00 0E FC 7F]

Stop Jet

CMD-ID	0017H
Description	stop jet
Send [<DATA>] format	Empty
Return [<DATA>] format	Empty

Example:

PC -> Printer: [7E 00 17 00 0C 00 00 00 00 00 00 00 00 00 52 F1 7F]

Printer -> PC: [7E 00 17 00 0C 00 06 00 00 00 00 00 00 00 9F A9 7F]

Start Print

CMD-ID	0018H
Description	Start print
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 18 00 0C 00 00 00 00 00 00 00 00 00 1E ED 7F]

Printer -> PC: [7E 00 18 00 0C 00 06 00 00 00 00 00 00 00 D3 B5 7F]

Stop Print

CMD-ID	0019H
Description	stop print
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

PC -> Printer:[7E 00 19 00 0C 00 00 00 00 00 00 00 00 00 00 8F B8 7F]

Printer -> PC: [7E 00 19 00 0C 00 06 00 00 00 00 00 00 00 42 E0 7F]

Trigger Print

CMD-ID	001AH
Description	Trigger print
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

PC -> Printer:[7E 00 1A 00 0C 00 00 00 00 00 00 00 00 00 00 3C 46 7F]

Printer -> PC: [7E 00 1A 00 0C 00 06 00 00 00 00 00 00 00 F1 1E 7F]

Set Date Time

CMD-ID	001BH
Description	Set date time
Send [<DATA>] format	20 bytes format is “yyyy.MM.dd-hh:mm:ss” for example “2017.06.30-17:30:00”
Return [<DATA>] format	empty

Example:

PC -> Printer: 7E 00 1B 00 0C 00 00 00 00 00 00 00 00 00 00 00

32 30 31 37 2E 30 36 2E 33 30 2D 31 37 3A 33 30 3A 30 30 00 67 44 7F]

Printer -> PC: [7E 00 1B 00 0C 00 06 00 00 00 00 00 00 60 4B 7F]

Get Date Time

CMD-ID	001CH
Description	Get date time
Send [<DATA>] format	empty
Return [<DATA>] format	20 bytes format is “yyyy.MM.dd-hh:mm:ss” for example “2017.06.30-17:30:00”

Example:

PC -> Printer: [7E 00 1C 00 0C 00 00 00 00 00 00 00 00 00 4B B3 7F]

Printer -> PC: [7E 00 1C 00 0C 00 06 00 00 00 00 00 00 00 00 00]

32 30 31 37 2E 30 36 2E 33 30 2D 31 37 3A 34 33 3A 33 39 00 09 D3 7F]

Get Font List

CMD-ID	001DH
Description	Get font list
Send [<DATA>] format	empty
Return [<DATA>] format	<p>[FontCount] 1 byte Number of fonts</p> <p>[FontNameList] An array of font names, each font name occupies 16 bytes</p>

Example:

PC -> Printer: [7E 00 1D 00 0C 00 00 00 00 00 00 00 00 00 00 DA E6 7F]

15

20 35 20 48 69 67 68 43 61 70 73 00 00 00 00 00

20 37 20 48 69 67 68 43 61 70 73 00 00 00 00 00

20 39 20 48 69 67 68 43 61 70 73 00 00 00 00 00

31 32 20 48 69 67 68 43 61 70 73 00 00 00 00 00

31 36 20 48 69 67 68 43 61 70 73 00 00 00 00 00

31 36 20 48 69 67 68 46 75 6C 6C 00 00 00 00 00 00

32 34 20 48 69 67 68 43 61 70 73 00 00 00 00 00

32 34 20 48 69 67 68 46 75 6C 6C 00 00 00 00 00 00

33 32 20 48 69 67 68 46 75 6C 6C 00 00 00 00 00 00

20 39 20 43 68 69 6E 65 73 65 00 00 00 00 00 00 00

31 32 20 43 68 69 6E 65 73 65 00 00 00 00 00 00

31 36 20 43 68 69 6E 65 73 65 00 00 00 00 00 00

32 34 20 43 68 69 6E 65 73 65 00 00 00 00 00 00 00

中国社会科学院文学研究所《文学评论》编辑部 2013 年 12 月

37 20 41 72 61 62 69 63 00 00 00 00 00 00 00 00 00 00

39 20 41 72 61 62 69 63 00 00 00 00 00 00 00 00 00 00

31 32 20 41 72 61 62 69 63 00 00 00 00 00 00 00 00 00

32 31 20 41 72 61 62 69 63 00 00 00 00 00 00 00 00 00

31 32 20 4B 6F 72 65 61 00 00 00 00 00 00 00 00 00 00

31 36 20 4B 6F 72 65 61 00 00 00 00 00 00 00 00 00 00

32 34 20 4B 6F 72 65 61 00 00 00 00 00 00 00 00 00 00

20 37 20 43 68 69 6E 65 73 65 00 00 00 00 00 00 00 00

63 FA 7F]

Get Message List

CMD-ID	001EH
Description	get message list
Send [<DATA>] format	empty
Return [<DATA>] format	[FileCount] 2 bytes Number of message [FileNameList] array of file names, each file name occupies 32 bytes

Example:

PC -> Printer: [7E 00 1E 00 0C 00 00 00 00 00 00 00 00 69 18 7F]

Printer -> PC: [7E 00 1E 00 0C 00 06 00 00 00 00 00 00 00 00 00 00

01 00 47 65 6E 53 74 64 5F 35 5F 31 2E 6E 6D 6B 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 A3 2C 7F]

Create Field (Text)

CMD-ID	001FH
Description	create field (Text)
Send [<DATA>] format	<p>[FieldType] 1 byte is fixed at 00h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Character rotation</p> <p> 1 no rotation</p> <p> 2 rotate 90 degrees clockwise</p> <p> 3 rotate 180 degrees clockwise</p> <p> 4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte character horizontal mirroring</p> <p>[MirrorY] 1 byte Character vertical mirroring</p> <p>[Revert] 1 byte Character reverse color</p> <p>[FontName] 16 bytes font name</p> <p>[Interval] 1 byte Character spacing</p> <p>[StrLength] 2 bytes string length</p> <p>[String] Uncertain length String content</p>
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 1F 00 0C 00 00 00 00 00 00 00 00 00 00 00 00 00]

00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 39 20 48 69 67 68 43 61 70 73 00 00 00 00 00 00 01

07 00 41 42 43 44 45 46 47 56 5F 7F]

Printer -> PC: [7E 00 1F 00 0C 00 06 00 00 00 00 00 00 00 35 15 7F]

Create Field (Barcode)

CMD-ID	001FH
Description	Create field(Barcode)
Send [<DATA>] format	<p>[FieldType] 1 byte fixed at 01h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirroring</p> <p>[MirrorY] 1 byte Vertical Mirroring</p> <p>[Revert] 1 byte Reverse color</p> <p>[Symbology] 1 byte barcode type</p> <p>[Option1] 1 byte</p> <p>[Option2] 1 byte</p> <p>[Option3] 1 byte</p> <p>[Reverse] 1 byte</p> <p>[StrLength] 2 bytes barcode content length</p> <p>[String] Uncertain length Barcode content</p>
Return [<DATA>] format	empty

Create Field (Logo)

CMD-ID	001FH
Description	create field(Logo)
Send [<DATA>] format	[FieldType] 1 byte fixed at 02h [PositionX] 2 bytes Horizontal coordinates [PositionY] 2 bytes Longitudinal coordinates [BoldX] 1 byte Horizontal bold [BoldY] 1 byte Vertical bold [Rotation] 1 byte Rotate 1 no rotation 2 rotate 90 degrees clockwise 3 rotate 180 degrees clockwise 4 rotate 270 degrees clockwise [MirrorX] 1 byte horizontal mirroring [MirrorY] 1 byte Vertical Mirroring [Revert] 1 byte Reverse color [Width] 2 bytes pattern width [Height] 2 bytes pattern height [Length] 2 bytes pattern content length [Data] Uncertain length Pattern content
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 1F 00 0C 00 00 00 00 00 00 00 00 00 00 00]

02 00 00 00 00 00 00 00 00 00 00 00 0A 00 0A 00 14 00 00 02 C0 03 70 00 4C 00 42 00
4C 00 58 00 60 00 C0 00 80 01 4A 82 7F]

Printer -> PC: [7E 00 1F 00 0C 00 06 00 00 00 00 00 00 00 35 15 7F]

Create Field (Remote Text)

CMD-ID	001FH
Description	Create field(Remote text)
Send [<DATA>] format	<p>[FieldType] 1 byte is fixed at 03h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirroring</p> <p>[MirrorY] 1 byte Vertical Mirroring</p> <p>[Revert] 1 byte Reverse color</p> <p>[FontName] 16 bytes font name</p> <p>[Interval] 1 byte Character spacing</p> <p>[CharCount] 2 bytes Number of characters</p>
Return [<DATA>] format	empty

Example:

03 00 00 00 00 00 00 00 00 00 00 20 35 20 48 69 67 68 43 61 70 73 00 00 00 00 00 01

0C 00 F5 06 7F]

Printer -> PC: [7E 00 1F 00 0C 00 06 00 00 00 00 00 00 00 35 15 7F]

Create Field (Remote Barcode)

CMD-ID	001FH
Description	Create field(Remote barcode)
Send [<DATA>] format	<p>[FieldType] 1 byte fixed at 04h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirroring</p> <p>[MirrorY] 1 byte Vertical Mirroring</p> <p>[Revert] 1 byte Reverse color</p> <p>[Symbology] 1 byte barcode type</p> <p>[Option1] 1 byte</p> <p>[Option2] 1 byte</p> <p>[Option3] 1 byte</p> <p>[Reverse] 1 byte</p> <p>[CharCount] 2 bytes Number of characters</p>
Return [<DATA>] format	empty

Create Field (DateTime Text)

CMD-ID	001FH
Description	Create field(date time text)
Send [<DATA>] format	<p>[FieldType] 1 byte fixed at 05h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirroring</p> <p>[MirrorY] 1 byte Vertical Mirroring</p> <p>[Revert] 1 byte Reverse color</p> <p>[Format] 20 bytes date format string</p> <p>Example: "%Y-%m-%d %H:%M:%S"</p> <p>[OffsetYear] 2 bytes year offset</p> <p>[OffsetMonth] 2 bytes Month offset</p> <p>[OffsetDay] 2 bytes day offset</p> <p>[OffsetHour] 2 bytes hour offset</p> <p>[OffsetMin] 2 bytes Minute offset</p> <p>[FontName] 16 bytes font name</p> <p>[Interval] 1 byte Character spacing</p> <p>[StrLength] 2 bytes The length of the string is fixed at 0000h</p>
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 1F 00 0C 00 00 00 00 00 00 00 00 00 00 00 00 00]

Printer -> PC: [7E 00 1F 00 0C 00 06 00 00 00 00 00 00 00 35 15 7F]

Create Field (DateTime Barcode)

CMD-ID	001FH
Description	Create field(date time barcode)
Send [<DATA>] format	<p>[FieldType] 1 byte fixed at 06h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirror</p> <p>[MirrorY] 1 byte Vertical Mirror</p> <p>[Revert] 1 byte Reverse color</p> <p>[Format] 20 bytes date format string</p> <p>Example: "%Y-%m-%d %H:%M:%S"</p> <p>[OffsetYear] 2 bytes year offset</p> <p>[OffsetMonth] 2 bytes Month offset</p> <p>[OffsetDay] 2 bytes day offset</p> <p>[OffsetHour] 2 bytes hour offset</p> <p>[OffsetMin] 2 bytes Minute offset</p> <p>[Symbology] 1 byte barcode type</p> <p>[Option1] 1 byte</p> <p>[Option2] 1 byte</p> <p>[Option3] 1 byte</p> <p>[Reverse] 1 byte</p> <p>[StrLength] 2 bytes The length of the barcode content is fixed as 0000h</p>
Return [<DATA>] format	empty

Create Field (SerialNum Text)

CMD-ID	001FH
Description	Create field (serial number text)
Send [<DATA>] format	<p>[FieldType] 1 byte fixed at 07h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirror</p> <p>[MirrorY] 1 byte Vertical Mirror</p> <p>[Revert] 1 byte Reverse color</p> <p>[Begin] 4 bytes Start value</p> <p>[End] 4 bytes End value</p> <p>[Step] 4 bytes Step value</p> <p>[Current] 4 bytes Current value</p> <p>[Repeats] 4 bytes Repeats</p> <p>[RepeatCount] 4 bytes Current repetitions</p> <p>[Hexadecimal] 1 byte</p> <p>[Digits] 1 byte Bit width</p> <p>[LeadingZero] 1 byte leading zero</p> <p>[FontName] 16 bytes font name</p> <p>[Interval] 1 byte Character spacing</p> <p>[StrLength] 2 bytes The length of the string is fixed as 0000h</p>
Return [<DATA>] format	empty

Example:

00 00 00 00 00 00 00 00 00 0A 04 01 20 39 20 48 69 67 68 43 61 70 73 00 00 00 00 00 00 00 00 00

00 C6 C0 7F]

Printer -> PC: [7E 00 1F 00 0C 00 06 00 00 00 00 00 00 35 15 7F]

Create Field (SerialNum Barcode)

CMD-ID	001FH
Description	Create field (serial number Barcode)
Send [<DATA>] format	<p>[FieldType] 1 byte is fixed at 08h</p> <p>[PositionX] 2 bytes Horizontal coordinates</p> <p>[PositionY] 2 bytes Longitudinal coordinates</p> <p>[BoldX] 1 byte Horizontal bold</p> <p>[BoldY] 1 byte Vertical bold</p> <p>[Rotation] 1 byte Rotate</p> <p>1 no rotation</p> <p>2 rotate 90 degrees clockwise</p> <p>3 rotate 180 degrees clockwise</p> <p>4 rotate 270 degrees clockwise</p> <p>[MirrorX] 1 byte horizontal mirror</p> <p>[MirrorY] 1 byte Vertical Mirror</p> <p>[Revert] 1 byte Reverse color</p> <p>[Begin] 4 bytes Start value</p> <p>[End] 4 bytes End value</p> <p>[Step] 4 bytes Step value</p> <p>[Current] 4 bytes Current value</p> <p>[Repeats] 4 bytes Repeats</p> <p>[RepeatCount] 4 bytes Current repetitions</p> <p>[Hexadecimal] 1 byte</p> <p>[Digits] 1 byte Bit width</p> <p>[LeadingZero] 1 byte leading zero</p> <p>[Symbology] 1 byte barcode type</p> <p>[Option1] 1 byte</p> <p>[Option2] 1 byte</p> <p>[Option3] 1 byte</p> <p>[Reverse] 1 byte</p> <p>[DataLength] 2 bytes fixed to 0000h</p>
Return [<DATA>] format	empty

Download Remote Buffer

CMD-ID	0020H
Description	Download Remote buffer
Send [<DATA>] format	[Length] 2 bytes string length [String] Uncertain length String content
Return [<DATA>] format	Section Indicates whether the remote segment buffer is full

Example:

0A 00 31 32 33 34 35 36 37 38 39 30 D4 50 7F]

Printer -> PC: [7E 00 20 00 0C 00 06 00 00 00 00 00 00 00 00 5F 20 7F]

Delete Last Field

CMD-ID	0021H
Description	Delete last field
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 21 00 0C 00 00 00 00 00 00 00 00 00 EA 97 7F]

Printer -> PC: [7E 00 21 00 0C 00 06 00 00 00 00 03 00 4F E5 7F]

Delete Message Content

CMD-ID	0022H
Description	Delete message content
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

PC -> Printer: [7E 00 22 00 0C 00 00 00 00 00 00 00 00 00 00 59 69 7F]

Printer -> PC: [7E 00 22 00 0C 00 06 00 00 00 00 00 00 94 31 7F]

Set Current Message

CMD-ID	0023H
Description	Set current message
Send [<DATA>] format	[FileName] File name, length 32 bytes, file name less than 32, filled with 0.
Return [<DATA>] format	empty

Example:

PC -> Printer: 7E 00 23 00 0C 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 A7 FA 7F]

Printer -> PC: [7E 00 23 00 0C 00 06 00 00 00 00 00 00 05 64 7F]

Set Aux Mode

CMD-ID	0024H
Description	Set Aux mode
Send [<DATA>] format	[Mode] 1 byte. 0 Turn off the auxiliary eye function 1 serial number reset 2 horizontal reversal 3 vertical reversal 4 horizontal and vertical reversal
Return [<DATA>] format	empty

Get Aux Mode

CMD-ID	0025H
Description	Get Aux mode
Send [<DATA>] format	empty
Return [<DATA>] format	[Mode] 1 byte. 0 Turn off the auxiliary eye function 1 serial number reset 2 horizontal reversal 3 vertical reversal 4 horizontal and vertical reversal

Set Reference Modulation

CMD-ID	0028H
Description	Set reference modulation
Send [<DATA>] format	[Mode] 1 byte., reference modulation value
Return [<DATA>] format	empty

Get Reference Modulation

CMD-ID	0029H
Description	Get reference modulation
Send [<DATA>] format	empty
Return [<DATA>] format	[Mode] 1 byte., reference modulation value

Reset Serial Number

CMD-ID	002AH
Description	Reset serial number
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Reset Count Length

CMD-ID	002BH
Description	Reset count length
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Print Trigger State

CMD-ID	1000H
Description	print trigger state
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

Printer -> PC: [7E 00 00 10 0C 00 00 00 00 00 00 00 00 00 F2 A3 7F]

Print Go State

CMD-ID	1001H
Description	Print go state
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

Printer -> PC: [7E 00 01 10 0C 00 00 00 00 00 00 00 00 00 A7 32 7F]

Print End State

CMD-ID	1002H
Description	Print end state
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

Printer -> PC: [7E 00 02 10 0C 00 00 00 00 00 00 00 00 00 59 81 7F]

Request Remote Data

CMD-ID	1003H
Description	Request remote data
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

Printer -> PC: [7E 00 03 10 0C 00 00 00 00 00 00 00 00 00 00 0C 10 7F]

Print Fault State

CMD-ID	1004H
Description	Print fault state
Send [<DATA>] format	empty
Return [<DATA>] format	empty

Example:

Printer -> PC: [7E 00 04 10 0C 00 00 00 00 00 00 00 00 00 00 00 AC F6 7F]

Communication Line

RS232 communication

For models that support RS232 communication, find the interface shown in Figure-1 and establish an RS232 communication line with the host as shown in Figure 2. The parameter settings such as the baud rate are shown in Figure-3.

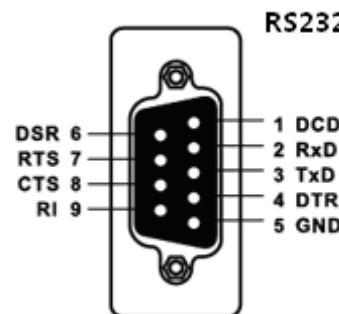


图-1

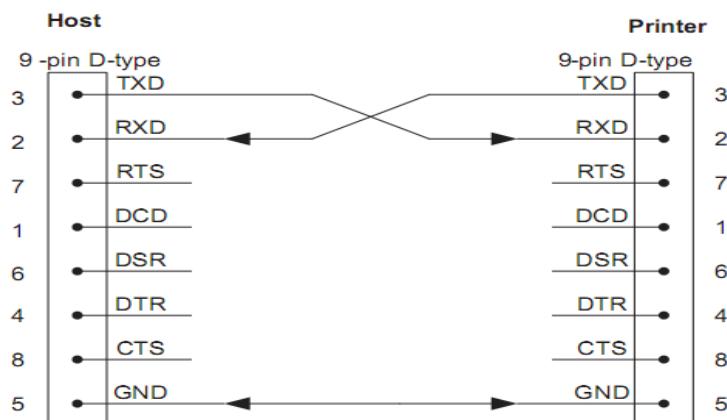


图-2



图-3