

# V 系列喷码机通讯协议

V1.0.0.20170823

# 目录

一、修订记录 .....	4
二、文档说明 .....	4
1 简介 .....	4
2 格式说明 .....	4
三、网络通讯说明 .....	5
1 基本描述。 .....	5
2 命令头和命令尾 .....	5
3 分隔符 .....	6
4 转义字符 .....	6
5 子命令 .....	6
6 命令列表 .....	7
6.1 返回指令 .....	7
6.2 系统指令 .....	7
6.3 喷印控制指令 .....	7
6.4 参数指令 .....	7
6.5 文件指令 .....	7
6.6 其他指令 .....	8
7 定长方案 .....	8
四、指令格式说明 .....	9
五、指令集 .....	9
1 返回指令 .....	9
1.1 正确返回 .....	9
1.2 错误返回 .....	9
2 系统指令 .....	10
2.1 获取设备基本信息 .....	10
2.2 获取系统状态指令 .....	11
2.3 获取卡匣信息 .....	13
2.4 修改设备名称 .....	14
3 喷印控制指令 .....	14

---

3.1 开启喷印 .....	14
3.2 停止喷印 .....	15
3.3 发送动态文本 .....	16
3.4 清除喷印缓存数据 .....	17
3.5 获取喷印状态 .....	17
4 喷印参数指令 .....	18
4.1 获取时间 .....	18
4.2 设置时间 .....	19
5 文件指令 .....	20
5.1 获取文件列表 .....	20
5.2 发送文件 .....	21
5.3 获取文件 .....	22
5.4 获取文件指定包 .....	23
6 系统功能指令 .....	25
6.1 获取产线速度 .....	25
6.2 设置产线速度 .....	25
六、附录 .....	26
1 校验码算法 .....	26
2 错误码表 .....	26

## 一 修订记录

版本	修订摘要	日期	修改者
1.0.0.20170522	添加喷印控制模块的协议指令	2017-5-22	Will
1.0.0.20170527	添加喷印参数指令、系统功能指令、系统信息指令和其他指令	2017-5-27	Benci
1.0.0.20170629	修改命令列表，修改指令格式说明（添加 SN 号），添加扫描设备、修改设备名称和获取墨盒信息 3 条指令	2017-6-29	Will
1.0.0.20170714	对整体的一些调整修改	2017-7-14	Bade
1.0.0.20170811	增加了部分获取、设置喷印参数指令，获取系统状态指令	2017-8-11	Benci
1.0.0.20170817	调整了函数分类，显示当前版本可用指令	2017-8-17	Ronnie
1.0.0.20170821	调整了发送和读取文件指令	2017-8-21	Ronnie

## 二 文档说明

### 1 简介

#### （1）通讯方式

设备支持通过以太网口，串口进行连接。客户必须保证设备和电脑的连接通畅。以太网口通过建立 TCP 或 UDP 进行通讯交互。

串口使用 RS232 进行通讯。

#### （2）通讯参数

UDP 设备通讯端口：18666

TCP 设备通讯端口：18888

串口设备通讯波特率：9600

### 2 格式说明

（1）绝大部分命令是明码指令，少数指令为二进制指令。

（2）明码指令中的识别标识为全部大写，如模板名称的识别标识为 MSGNAME。

（3）示例中的机器的 SN 号统一为 12345679。

## 三 网络通讯说明

### 1 基本描述。

- (1) 一个完整命令由命令头，命令尾和中间数据部分构成；
- (2) 整条命令由可见字符组成；
- (3) 数据部分可以是由多条子命令组成；
- (4) 子命令之间以及子命令内部由分隔符隔开；



### 2 命令头和命令尾

整条命令是由可见字符组成,所以为了避免与数据内容重复造成误读,命令头尾将由使用频率最低的字母加上字符组合而成。

组合字符	使用位置	描述
>BON>	头	Begin of No check code Command 无校验码命令
<BON<	头	Begin of Device Send Command 设备主动发送命令
=EOC=	尾	End of Command 命令尾

### 3 分隔符

设定了三级的分隔符，如下：

- 主命令分隔符[ ] ；
- 明码子命令分隔符[ ^ ] ； ；
- 子命令内参数分隔符[ ` ] [ `` ]

[ ` ]后为普通明码数据

[ `` ]后为二进制流数据

### 4 转义字符

明码指令中，由于分隔符也是可视字符，因此在数据中需要用到这三个字符的时候，需要用到转义字符 \，比如：

- a) A|B ==> A\|B
- b) A^B ==> A\^B
- c) A`B ==> A\`B
- d) A||B ==> A\\|B

二进制码中无需添加转义字符。

### 5 子命令



注：这里的分隔符没有表示出

## 6 命令列表

### 6.1 返回指令

序号	命令	明码命令	说明
1	正确返回	CMD_OK	正确返回
2	错误返回	CMD_ERROR	错误返回

### 6.2 系统指令

序号	命令	标识	说明
1	系统基本信息指令	CMD_BASEINFO	喷码机基本信息，如版本号 IP 等
2	系统状态指令	CMD_SYSSTATUS	
3	卡匣信息	CMD_INKINFO	
4	修改设备名称	CMD_CHANGEDEVICENAME	

### 6.3 喷印控制指令

序号	命令	标识	说明
1	开启喷印	CMD_PRINTON	
2	停止喷印	CMD_PRINTOFF	
3	发送动态文本	CMD_DYNTXT	
4	清除喷印缓存	CMD_CLEANCACHE	
5	获取喷印状态	CMD_PRINTSTATUS	

### 6.4 参数指令

序号	命令	标识	说明
1	获取时间	CMD_GETDATETIME	
2	设置时间	CMD_SETDATETIME	

### 6.5 文件指令

序号	命令	明码标识	说明
1	获取文件列表	CMD_GETFILESList	获取指定文件类型列表
2	下载指定文件	CMD_DOWNLOADFILE	发送指定文件

3	上传指定文件	CMD_UPLOADFILE	获取指定文件

序号	命令	明码名称	说明
1	获取产线速度	CMD_GETLINESPEED	
2	设置产线速度	CMD_SETLINESPEED	

- 数字部分，前面补 0，如 3.6 => 003.6
- 字符串部分，使用符号~将内容括起来，我们只读取两个符号之间的内容，所以可以添加更多的~进行占位。内部需要用到~的，也使用转义字符。

**"** [Icons: folder, document, document, document, calendar, hourglass, printer, mouse, camera, folder] **"**

**"** [Icons: folder, document, document, document] **"""""""""""**



## 四 指令格式说明

指令 = 指令头 + SN 号 + 具体指令数据 + 指令尾

BEGIN + SN + CMD + END

	项目	大小	说明	备注
1	BEGIN	5	指令头	>BON>: 上位机发送无校验码指令 <BON<: 设备发送无校验码指令
2	SN	4	设备 SN 号	如果为 UDP 广播则为 0
3	DATA	1...N	具体指令数据	由指令类型和输入参数构成，具体查看指令集
4	END	5	指令尾	=EOC=: 指令尾

## 五 指令集

### 1 返回指令

#### 1.1 正确返回

- 1 指令码: CMD\_OK
- 2 功能说明: 返回的命令字符串中表示指令成功执行的指令。

#### 1.2 错误返回

- 1 指令码: CMD\_ERROR
- 2 功能说明: 返回的命令字符串中表示指令未能成功执行的指令。

## 2 系统指令

### 2.1 获取设备基本信息

- 1 指令码：CMD\_BASEINFO
- 2 指令功能：获取设备的基本信息，包括版本号，IP 地址等
- 3 数据包：

#### 3.1 发送数据包

有 2 种格式

##### 3.1.1 获取全部基本信息

项目	类型	说明
CMD_BASEINFO	字符串	不指定表示代表获取设备的全部基本信息

##### 3.1.2 获取指定信息

项目	类型	说明
CMD_BASEINFO	字符串	
要获取设备基本信息的标识 1	字符串	要获取设备基本信息的标识 1
...	字符串	...
要获取设备基本信息的标识 n	字符串	要获取设备基本信息的标识 n

#### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD_BASEINFO	字符串	
要获取设备基本信息的标识 1	字符串	要获取设备基本信息的标识 1
标识 1 的值	字符串	标识 1 的值
...	字符串	...
要获取设备基本信息的标识 n	字符串	要获取设备基本信息的标识 n
标识 n 的值	字符串	标识 n 的值

#### 3.3 标识说明：

标识可按要求组合，具体标识如下：

标识	说明
SOFTV	软件版本
HARDV	硬件版本
DEVSN	设备 SN 号

CUSCD	客户编码
IPADR	IP 地址
SUBMK	子网掩码
DEFGY	网关地址
MACADR	MAC 地址
PTCLV	Protocol 版本号
MODEL	V1H(设备机型)
	V1
	V2

### 3.4 数据包示例:

发送:

>BON>|12345679|1^CMD\_BASEINFO|=EOC= （要求获取设备的基本信息）

>BON>|12345679|1^ BASEINFO `DEVSIN`IPADR|=EOC= （要求获取设备的 SN 号和 IP 地址）

接收:

<BON<|12345679|1^CMD\_OK`CMD\_BASEINFO`DEVSIN`201711`IPADR`192.168.0.111|=EOC=  
（获取到设备的 SN 号和 IP 地址）

<BON<|12345679|1^CMD\_ERROR`CMD\_BASEINFO|=EOC= （无法获取到设备信息）

## 2.2 获取系统状态指令

(1) 指令码: CMD\_SYSSTATUS

(2) 指令功能: 获取设备系统的状态信息

(3) 发送数据包

项目	类型	说明
CMD	字符串	指令码: CMD_SYSSTATUS
标识 1	字符串	要获取的标识 1
标识 1 的值	字符串	要获取的标识 1 的值
...	字符串	要获取的标识 2~n-1
标识 n	字符串	要获取的标识 n
标识 n 的值	字符串	要获取的标识 n 的值

(4) 接收数据包

项目	长度	类型	说明
RETURN	1	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	1	字符串	指令码: CMD_SYSSTATUS

(5) 标识说明：（标识可按要求组合）

标识	说明
<b>SYSSTATUS</b>	返回一些系统状态信息
<b>USBSTATUS</b>	USB 连接设备情况
<b>ENCODER</b>	同步器状态
<b>PHOTOCELL</b>	电眼状态

SYSSTATUS: 此值为系统状态。获取此项后接一个参数。

USBSTATUS: 此值为 USB 状态，获取此项后接 2 个参数。“USB 序列号” “未连接” 或 “U 盘”

ENCODER: 此值为同步器状态。获取此项后接一个参数。

PHOTOCELL: 此值为电眼类型。获取此项后接一个参数。

(6) 数据包示例:

发送:

>BON>|12345679|1^CMD\_SYSSTATUS`USBSTATUS|=EOC= （获取 USB 状态）

成功返回:

<BON<|12345679|1^CMD\_OK`CMD\_SYSSTATUS`USBSTATUS`OFF|=EOC=

（成功获取 USB 状态）

失败返回:

<BON<|12345679|1^CMD\_ERROR`CMD\_SYSSTATUS |=EOC= （获取失败）

(7) 标识返回参数说明:

标识	参数
<b>SYSSTATUS</b>	PRINTINGMSG 正在喷印的资料，没有则返回 NULL
	DIRECTION L2R （左到右） R2L （右到左）
	NOZZLE LEFT （左喷） RIGHT （右喷） DOUBLE （双喷）
	MIRROR NONE （关闭） HORIZONTAL （水平镜像） VERTICAL （垂直镜像）
	DPI 解析度
	CACHE 缓存量
	TIMES 连喷模式：连喷次数 往返模式：往返次数
	INTERVAL 连喷模式：连喷间隔 往返模式：反向延迟
	OUTPUT 产量值
	PREPURGE OFF LOW MEDIUM HIGH CUSTOM
	PREPURGEMODE DOUBLE （双喷嘴）

		WORKING (工作喷嘴)
USBSTATUS	NUM (USB 序号)	OFF (关)
		DISK (U 盘)
PHOTOCELL	INTERNAL (内置电眼)	
	EXTERNAL (外置电眼)	
ENCODER	OFF(未启用同步器)	
	ON (启用同步器)	

注：(因精度原因 INTERVAL 的传递数值放大 1000 倍)

## 2.3 获取卡匣信息

- 1 指令码：CMD\_INKINFO
- 2 指令功能：获取卡匣油墨信息
- 3 指令包：

### 3.1 发送数据包

项目	类型	说明
CMD	字符串	明码指令码：CMD_INKINFO
要获取的内容标识	字符串	例如“PERCENT”

### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	指令码：CMD_INKINFO
油墨卡匣数量	字符串	只有一个喷头则固定为 1
要获取卡匣信息的标识 1	字符串	要获取卡匣信息的标识 1
标识 1 的值	字符串	标识 1 的值
...	...	...

### 3.3 数据包示例

发送：

>BON>|12345679|1^CMD\_INKINFO`VOLUME |=EOC=

(表示含义：获取油墨含量)

接收：

<BON<|12345679|1^CMD\_OK`CMD\_INKINFO`1` VOLUME`42ML|=EOC=

(油墨含量为 42 毫

升)

<BON<|12345679|1^CMD\_ERROR`CMD\_INKINFO`NULL|=EOC=

### 3.4 标识

标识	说明
VOLUME	油墨含量
TYPE	油墨类型
INKSN	油墨序列号
PROSPECTOUTPUT	当前资料预期可喷印笔数
REMAININGOUTPUT	剩余可喷印笔数

PERCENTVOLUME	油墨剩余量百分比数值
CUSTOMCODE	客户码
STATUS	油墨状态

错误	说明
NULL	没有找到油墨卡匣
UNAVAIL	不可用

## 2.4 修改设备名称

- 1 指令码: CMD\_CHANGEDEVICENAME
- 2 指令功能: 修改设备名称
- 3 指令包:

### 3.1 发送数据包

项目	类型	说明
CMD	字符串	明码指令码: CMD_CHANGEDEVICENAME
要更改的名称	字符串	例如“PRINTER 1”

### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_CHANGDEVICENAME

### 3.3 数据包示例

发送:

```
>BON>|12345679|1^CMD_CHANGEDEVICENAME`PRINTER1`|=EOC=
```

（表示含义: 修改设备名称为 PRINTER1）

接收:

```
<BON<|12345679|1^CMD_OK`CMD_CHANGEDEVICENAME|=EOC= (修改成功)
```

## 3 喷印控制指令

### 3.1 开启喷印

- 4 指令码: CMD\_PRINTON
- 5 指令功能: 根据指定的资料模板名称, 开启喷印
- 6 指令包:

## 6.1 发送数据包

项目	类型	说明
CMD	字符串	明码指令码: CMD_PRINTON
要喷印的资料名	字符串	例如“MSG001”

## 6.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_PRINTON
ERROR	字符串	错误码

错误码	说明
LOWPOWER	电量过低
INKDISCONNECT	卡匣未连接
INKUNAVAILABLE	卡匣无效
SYSTEMERROR	系统错误
MESSAGENOFIND	资料未找到
INPRINTING	正在喷印中

## 6.3 数据包示例

发送:

>BON>|12345679|1^CMD\_PRINTON`MSG001|=EOC=

(表示含义: 开始喷印, 喷印的资料为“MSG001”)

接收:

<BON<|12345679|1^CMD\_OK`CMD\_PRINTON|=EOC= (开启喷印成功)

<BON<|12345679|1^CMD\_ERROR`CMD\_PRINTON`LOWPOWER|=EOC=  
(电量过低无法开启喷印)

## 3.2 停止喷印

- 1 指令码: CMD\_PRINTOFF
- 2 指令功能: 关闭正在喷印的模板
- 3 指令包:

### 3.1 发送数据包

项目	类型	说明
CMD	字符串	明码指令码: CMD_PRINTOFF

### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_PRINTOFF

### 3.3 数据包示例:

发送：

>BON>|12345679|1^CMD\_PRINTOFF|=EOC=

成功返回：

<BON<|12345679|1^CMD\_OK^CMD\_PRINTOFF|=EOC=

失败返回：

<BON<|12345679|1^CMD\_ERROR^CMD\_PRINTOFF|=EOC=

### 3.3 发送动态文本

- 1 指令码：CMD\_DYNTEXT
- 2 指令功能：发送动态文本数据给喷码机
- 3 指令包：

#### 3.1 发送数据包

项目	类型	说明
CMD	字符串	指令码：CMD_DYNTEXT
动态源的个数	字符串	动态源的个数
动态源 1 的名称	字符串	动态源 1 的名称
动态源 2 的名称	字符串	动态源 2 的名称
...	字符串	其他的动态源的名称
动态源 1 的第一笔数据	字符串	动态源 1 的第一笔数据
动态源 2 的第一笔数据	字符串	动态源 2 的第一笔数据
...	字符串	其他的动态源的第一笔数据
动态源 1 的第二笔数据	字符串	动态源 1 的第二笔数据
动态源 2 的第二笔数据	字符串	动态源 2 的第二笔数据
...	字符串	其他的动态源的第二笔数据
...	字符串	其他笔数据

#### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	指令码：CMD_DYNTEXT

#### 3.3 数据包示例：

发送：

- 1) 给动态数据 DynamicText1 发送 aaaa,bbbb,cccc 三笔数据

>BON>|12345679|1^CMD\_DYNTEXT`1`DynamicText1`aaaa`bbbb`cccc|=EOC=

- 2) 给动态数据 DynamicText1, DynamicText2, DynamicText3 发送二笔数据

>BON>|12345679|1^DYNTEXT`3`DynamicText1`DynamicText2`DynamicText3`aaaa`bbbb`cccc`aaa11`b



bbb22`cccc32|=EOC=

成功返回：

<BON<|12345679|1^CMD\_OK`CMD\_DYNTEXT|=EOC=

失败返回：

<BON<|12345679|1^CMD\_ERROR`CMD\_DYNTEXT|=EOC=

### 3.4 错误码

错误码	说明
NOPRINTING	不在喷印中
NODYNAMICTEXT	没有动态文本对象

## 3.4 清除喷印缓存数据

- 1 指令码：CMD\_CLEANCACHE
- 2 指令功能：清除喷码机中的喷印缓存数据
- 3 指令包：

### 3.1 发送数据包

项目	类型	说明
CMD	字符串	明码指令码：CMD_CLEANCACHE

### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	指令码：CMD_CLEANCACHE

### 3.3 数据包示例

发送：

>BON>|12345679|1^CMD\_CLEANCACHE|=EOC=

成功返回：

<BON<|12345679|1^CMD\_OK`CMD\_CLEANCACHE|=EOC=

失败返回：

<BON<|12345679|1^CMD\_ERROR`CMD\_CLEANCACHE|=EOC=

## 3.5 获取喷印状态

- 1 指令码：CMD\_PRINTSTATUS
- 2 指令功能：获取当前是否开启喷印，当前喷印资料名称等信息
- 3 数据包：

### 3.1 发送数据包:

项目	类型	说明
CMD	字符串	明码指令码: CMD_PRINTSTATUS

### 3.2 接收数据包:

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_PRINTSTATUS

### 3.3 标识:

标识	说明
ISPRINTING	是否开启喷印
PRINTINGMSG	正在喷印的资料名称, 未开启喷印则获取到 NULL 字符串
PRODUCTCOUNTER	产量计数器

### 3.4 数据包示例:

发送:

>BON>|12345679|1^CMD\_PRINTSTATUS`PRINTINGMSG`PRODUCTCOUNTER|=EOC=

接收:

<BON<|12345679|1^CMD\_OK`CMD\_PRINTSTATUS`PRINTINGMSG`MSG001`  
PRODUCTCOUNTER`6|=EOC=

## 4 喷印参数指令

### 4.1 获取时间

1. 指令码: CMD\_GETTIME
2. 指令包:
3. 发送数据包

项目	类型	说明
CMD	字符串	指令码: CMD_TIME
标识 1	字符串	要获取的标识 1
...	字符串	要获取的标识 2~n-1
标识 n	字符串	要获取的标识 n

### 4. 接收数据包

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_TIME

标识 1	字符串	要设置的标识 1
标识 1 的值	字符串	要设置的标识 1 的值
...	字符串	要设置的标识 2~(n-1)
标识 n	字符串	要设置的标识 n
标识 n 的值	字符串	要设置的标识 n 的值

5. 标识说明：（标识可按需要组合）

**DATETIME**：时间。

6. 数据包示例：

发送：>BON>|12345679|1^CMD\_GETTIME`DATETIME |=EOC= （获取系统时间）

成功返回：

<BON<|12345679|1^CMD\_OK`CMD\_GETTIME`DATETIME`NUM1`NUM2`NUM3`NUM4`NUM5`NUM6|=EOC= （成功返回系统时间）

失败返回：<BON<|12345679|1^CMD\_ERROR`CMD\_GETTIME|=EOC= （获取失败）

（7）标识参数说明：

标识	参数 1	参数 2	参数 3	参数 4	参数 5	参数 6
<b>DATETIME</b>	NUM1（年）	NUM2（月）	NUM3（日）	NUM4（时）	NUM5（分）	NUM6（秒）

## 4.2 设置时间

1. 指令码：CMD\_SETTIME

2. 指令包：

3. 发送数据包

项目	长度	类型	说明
<b>CMD</b>	3	字符串	指令码：CMD_SETTIME
标识 1	N	字符串	要设置的标识 1
标识 1 的值	N	字符串	要设置的标识 1 的值
...	N	字符串	要设置的标识 2~n-1
标识 n	N	字符串	要设置的标识 n
标识 n 的值	N	字符串	要设置的标识 n 的值

4. 接收数据包

项目	长度	类型	说明
<b>RETURN</b>	1	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
<b>CMD</b>	1	字符串	指令码：CMD_SETTIME

5. 标识说明：（标识可按需要组合）

**DATETIME**：时间。

## 6. 数据包示例：

发送：

>BON>|12345679|1^CMD\_SETTIME`DATETIME`NUM1`NUM2`NUM3`NUM4`NUM5`NUM6

|=EOC=（获取系统时间）

成功返回：<BON<|12345679|1^CMD\_OK`CMD\_SETTIME`|=EOC=（成功设置系统时间）

失败返回：<BON<|12345679|1^CMD\_ERROR`CMD\_SETTIME`ERROR|=EOC=（设置失败）

## 7. 标识参数说明：

标识	参数 1	参数 2	参数 3	参数 4	参数 5	参数 6
DATETIME	NUM1（年）	NUM2（月）	NUM3（日）	NUM4（时）	NUM5（分）	NUM6（秒）

# 5 文件指令

## 5.1 获取文件列表

1 指令码：CMD\_GETFILESLIST

2 指令功能：获取指定类型文件的列表

3 数据包：

### 3.1 发送数据包

项目	类型	说明
CMD	字符串	明码指令码：CMD_GETFILESLIST
要获取的文件标识	字符串	资料：MSG； 图片：LOGO； 字体：FONT

### 3.2 接收数据包

项目	类型	说明
RETURN	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	指令码：CMD_GETFILESLIST
要获取的文件标识 1	字符串	MSG,LOGO,FONT
总数量 1	字符串	文件标识 1 对应的文件总数量
文件名称 1	字符串	文件标识 1 的第 1 个文件名称
...	...	...
文件名称 N	字符串	文件标识 1 的第 N 个文件名称
要获取的文件标识 2	字符串	MSG,LOGO,FONT
总数量 2	字符串	文件标识 2 对应的文件总数量
文件名称 1	字符串	文件标识 2 的第 1 个文件名称
...	...	...
文件名称 N	字符串	文件标识 2 的第 N 个文件名称
...	...	...

### 3.3 示例

发送:

```
>BON>|12345679|1^CMD_GETFILESLIST`LOGO|=EOC=
```

成功返回:

```
<BON<|12345679|1^101`501`LOGO`5`Logo1`Logo2`Logo3`Logo4`Logo5|=EOC=
```

失败返回:

```
<BON<|12345679|1^CMD_ERROR`CMD_GETFILESLIST|=EOC=
```

## 5.2 发送文件

- 1 指令码: CMD\_DOWNLOADFILE
- 2 指令功能: 将资料文件发送到设备上
- 3 数据包:

### 3.1 发送资料包

项目	类型	说明
CMD	字符串	CMD_DOWNLOADFILE
文件数量	字符串	要发送的文件数量
文件 1 的名称	字符串	文件 1 的名称(带文件扩展名)
文件 1 的长度	字符串	文件 1 的文件大小
文件 1 的类型	字符串	文件 1 的类型（类型详见类型标识说明）
文件 1 的所属目录	字符串	目前只有资料文件需要提供资料名称，其他类型文件为 NULL
文件 1 的总包数	字符串	文件 1 的分包数(当文件 1 内容超出 5K 后需要分包)
文件 1 的分包索引号	字符串	文件 1 的分包索引号 (索引号由 1 开始; 总包数为 1 时, 索引号也为 1,)
...	...	...
文件 N 的名称	字符串	文件 N 的名称
文件 N 的类型	字符串	文件 N 的类型（类型详见类型标识说明）
文件 N 的所属目录	字符串	目前只有资料文件需要提供资料名称，其他类型文件为 NULL
文件 N 的总包数	字符串	文件 N 的分包数(当文件 N 内容超出 5K 后需要分包)
文件 N 的分包索引号	字符串	文件 N 的分包索引号（如总包数为 1 时, 索引号也为 1）
文件 1 本包数据长度	字符串	文件 1 本数据包中的数据长度
文件 1 的二进制数据	数据流	文件 1 的具体数据
...	...	...
文件 N 本包数据长度	字符串	文件 N 本数据包中的数据长度
文件 N 的二进制数据	数据流	文件 N 的具体数据

### 3.2 接收资料包

项目	类型	说明
RETURN	字符串	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	指令码：CMD_DOWNLOADFILE

### 3.3 标识说明：

文件类型	说明
MSG	资料文件
LOGO	图片文件
FONT	字体文件
UPGRADE	升级包文件

### 3.4 数据包示例：

发送：

>BON>|12345679|1^CMD\_DOWNLOADMSG`3`History.ini`86`MSG`MSG001`1`1`  
Page.ini`34`MSG`MSG001`1`1`Setup.ini`238`MSG`MSG001`1`1`86`History.ini 二进制数据  
`34`Page.ini 二进制数据`238`Setup.ini 二进制数据|=EOC=

接收：

<BON<|12345679|1^CMD\_OK`CMD\_DOWNLOADMSG|=EOC= （发送资料文件成功）  
<BON<|12345679|1^CMD\_ERROR`CMD\_DOWNLOADMSG|=EOC= （发送资料文件

失败）

## 5.3 获取文件

#### 1. 指令码：CMD\_UPLOADFILE

#### 2. 指令包：从设备获取文件，需要配合 CMD\_UPLOADFILEPACK 一起取得文件数据

#### 3. 发送数据包

项目	类型	说明
CMD	字符串	指令码：CMD_UPLOADFILE
文件数量	字符串	要获取的文件总数量
文件名称	字符串	要获取的文件名称
文件类型	字符串	要获取的文件类型标识

#### 4. 接收数据包

返回将要接收文件的信息

项目	类型	说明
RETURN	数值	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	CMD_UPLOADFILE

文件总数量	字符串	将要返回的文件总数量
文件 1 的名称	字符串	文件 1 的名称
文件 1 的长度	字符串	文件 1 的文件大小
文件 1 的类型	字符串	文件 1 的类型（类型详见类型标识说明）
文件 1 的所属目录	字符串	资料文件这里是资料名称，其他类型文件为 NULL
文件 1 的总包数	字符串	文件 1 的分包数(当文件 1 内容超出 5K 后需要分包)
...	...	...
文件 N 的名称	字符串	文件 N 的名称
文件 N 的类型	字符串	文件 N 的类型（类型详见类型标识说明）
文件 N 的所属目录	字符串	目前只有资料文件需要提供资料名称，其他类型文件为 NULL
文件 N 的总包数	字符串	文件 N 的分包数(当文件 N 内容超出 5K 后需要分包)

#### 5. 数据包示例：

发送：>BON>|12345679|1^CMD\_UPLOADFILE`1`MSG001`MSG|=EOC=（获取资料 MSG001）

接收：

<BON<|12345679|1^CMD\_OK`CMD\_UPLOADFILE`3`History.ini`86`MSG`MSG001`1`Page.ini`34`MSG`MSG001`1`Setup.ini`6000`MSG`MSG001`2|=EOC=（History.ini 有一个包，Page.ini 有 1 个包，Setup.ini 有 2 个包）

<BON<|12345679|CMD\_ERROR`CMD\_UPLOADMSG|=EOC=（获取资料文件失败）

## 5.4 获取文件指定包

#### 1. 指令码：CMD\_UPLOADFILEPACKAGE

#### 2. 指令包：从设备获取指定的文件包，需配合 CMD\_UPLOADFILE 一起使用

#### 3. 发送数据包

项目	类型	说明
CMD	字符串	指令码：CMD_UPLOADFILEPACKAGE
文件数量	字符串	要获取的文件总数量
文件名称	字符串	要获取的文件名称
文件类型	字符串	要获取的文件类型标识
文件包索引号	字符串	要获取的指定文件的文件包

#### 4. 返回数据包

项目	类型	说明
RETURN	数值	成功返回：CMD_OK 失败返回：CMD_ERROR
CMD	字符串	CMD_UPLOADFILEPACKAGE
文件数量	字符串	本次获取的文件数量
文件 1 的名称	字符串	文件 1 的名称
文件 1 的长度	字符串	文件 1 的文件大小
文件 1 的类型	字符串	文件 1 的类型（类型详见类型标识说明）
文件 1 的所属目录	字符串	资料文件这里是资料名称，其他类型文件为 NULL
文件 1 的总包数	字符串	文件 1 的分包数(当文件 1 内容超出 5K 后需要分包)

文件 1 的分包索引号	字符串	文件 1 的分包索引号（如总包数为 1 时，索引号也为 1）
...	...	...
文件 N 的名称	字符串	文件 N 的名称
文件 N 的类型	字符串	文件 N 的类型（类型详见类型标识说明）
文件 N 的所属目录	字符串	目前只有资料文件需要提供资料名称，其他类型文件为 NULL
文件 N 的总包数	字符串	文件 N 的分包数(当文件 N 内容超出 5K 后需要分包)
文件 N 的分包索引号	字符串	文件 N 的分包索引号（如总包数为 1 时，索引号也为 1）
文件 1 本包数据长度	字符串	文件 1 本数据包中的数据长度
文件 1 的二进制数据	数据流	文件 1 的具体数据
...	...	...
文件 N 本包数据长度	字符串	文件 N 本数据包中的数据长度
文件 N 的二进制数据	数据流	文件 N 的具体数据

## 6. 数据包示例：

发送：

>BON>|12345679|1^CMD\_UPLOADFILEPACKAGE`1` History.ini`MSG`MSG001`1|=EOC=  
(获取资料 MSG001 中的 History.ini)

接收：

<BON<|12345679|1^CMD\_OK`CMD\_UPLOADFILEPACKAGE`1` History.ini`86`MSG`MSG001`1`1`86`

History.ini 二进制数据|=EOC=

(获取资料文件 MSG001 中 History.ini)

发送：

>BON>|12345679|1^CMD\_UPLOADFILEPACKAGE`1` Setup.ini`MSG`MSG001`1|=EOC=  
(获取资料 MSG001 中的 Setup.ini 的第一个包)

接收：

<BON<|12345679|1^CMD\_OK`CMD\_UPLOADFILEPACKAGE`1` Setup.ini`6000`MSG`MSG001`2`1`5120`

Setup.ini 二进制数据|=EOC= (Setup.ini 的第一个包 5K=5\*1024=5120 字节)

发送：

>BON>|12345679|1^CMD\_UPLOADFILEPACKAGE`1` Setup.ini`MSG`MSG001`2|=EOC=  
(获取资料 MSG001 中的 Setup.ini 的第二个包)

接收：

<BON<|12345679|1^CMD\_OK`CMD\_UPLOADFILEPACKAGE`1` Setup.ini`6000`MSG`MSG001`2`2`880`

Setup.ini 二进制数据|=EOC= (Setup.ini 的第二个包)

发送：

<BON<|12345679|1^CMD\_UPLOADFILEPACKAGE`1` Page.ini`MSG`MSG001`1|=EOC=

接收：

<BON<|12345679|1^CMD\_OK`CMD\_UPLOADFILEPACKAGE`1` Page.ini`34`MSG`MSG001`1`1`34`

Page.ini 二进制数据|=EOC= (Page.ini)

失败返回：

<BON<|12345679|CMD\_ERROR`CMD\_UPLOADMSG|=EOC= (获取资料文件失败)



## 6 系统功能指令

### 6.1 获取产线速度

1. 指令码: CMD\_GETLINESPEED
2. 指令包: 获取设备上设置的产线速度
3. 发送数据包

项目	类型	说明
CMD	字符串	指令码: CMD_GETLINESPEED

4. 接收数据包

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_GETLINESPEED

5. 数据包示例:

发送: >BON>|12345679|1^CMD\_GETLINESPEED |=EOC=

成功返回:

<BON<|12345679|1^CMD\_OK`CMD\_GETLINESPEED`30.0 |=EOC= (成功获取)

失败返回:

<BON<|12345679|1^CMD\_ERROR`CMD\_GETLINESPEED |=EOC= (重置失败)

### 6.2 设置产线速度

1. 指令码: CMD\_SETLINESPEED
2. 指令包: 设置设备上设置的产线速度
3. 发送数据包

项目	类型	说明
CMD	字符串	指令码: CMD_SETLINESPEED
产线速度的值	字符串	产线速度

4. 接收数据包

项目	类型	说明
RETURN	字符串	成功返回: CMD_OK 失败返回: CMD_ERROR
CMD	字符串	指令码: CMD_SETLINESPEED

5. 数据包示例:

发送: >BON>|12345679|1^CMD\_SETLINESPEED`20.0 |=EOC=

成功返回:

<BON<|12345679|1^CMD\_OK`CMD\_SETLINESPEED |=EOC= （成功设置）

失败返回：

<BON<|12345679|1^CMD\_ERROR`CMD\_SETLINESPEED |=EOC= （设置失败）

## 六 附录

### 1 校验码算法

### 2 错误码表

```
#define SysSetup_Reset_message "MESSAGE"
#define SysSetup_Reset_image "IMAGE"
#define SysSetup_Reset_printpara "PRTPARA"
#define SysSetup_Reset_syspara "SYSPARA"
#define SysSetup_Reset_language "LANGUAGE"
#define SysSetup_Reset_font "FONT"
#define ToolsSend_Upgrade 601
#define ToolsSend_Upgrade_type "type"
#define ToolsSend_Upgrade_updatefile "updatefile"
#define ToolsSend_RestoreandBackup "602"
#define ToolsSend_RestoreandBackup_backup "backup"
#define ToolsSend_RestoreandBackup_type "type"
#define ToolsSend_RestoreandBackup_path "path"
#define ToolsSend_RestoreandBackup_restore "restore"
#define ToolsSend_RestoreandBackup_datatype "datatype"
#define ToolsSend_RestoreandBackup_imagetype "imagetype"
#define ToolsSend_RestoreandBackup_printparatype "printparatype"
#define ToolsSend_RestoreandBackup_sysparatype "sysparatype"
#define ToolsSend_RestoreandBackup_datafile "datafile"
#define ToolsSend_RestoreandBackup_imagefile "imagefile"
#define ToolsSend_RestoreandBackup_printparafilename "printparafilename"
#define ToolsSend_RestoreandBackup_sysparafilename "sysparafilename"
#define ToolsSend_RestoreandBackup_data "data"
#define ToolsSend_RestoreandBackup_Image "Image"
#define ToolsSend_RestoreandBackup_Printpara "Printpara"
#define ToolsSend_RestoreandBackup_syspara "syspara"
#define ToolsSend_ProductLine 603
#define ToolsSend_length "length"
#define ToolsSend_statu "statu"
#define ToolsSend_speed "speed"
#endif // SJ_PROTOCOLDEFINED_H
```