Simplified JetFileII Protocol Ver3.4

Version Updated Record

Version No.	Notes	Revised By	Date Revised
Ver1.0	First Version	YN	2006-03-10
Ver2.0	Add commands demonstration	CZP	2006-05-17
Ver2.1	Add command <0x06> Add commands demonstration for writing file to RAM drive & to Flash drive	CZP	2006-08-27
Ver2.2	Revised command format Add <0xXX> for all hex commands	YN	2006-09-07
Ver2.3	Revised font control commands	Soar	2007-01-24
Ver2.4	Explain of commands Prefix	Soar	2007-07-26
Ver2.5	Add a new format, support and test it	Soar	2007-10-11
Ver3.0	Typeset again Add list control command	Lei	2008-11-08
Ver3.1	Modify and check control commands	Lei	2009-01-12
Ver3.2	Writing simple BMP pictures	Lei	2009-04-24
Ver3.3	Add the code for the front Reserve 1 to 9 Change the translation of the Chinese sample.	Sureone	2010-5-6
Ver3.4	Add a sample to write a BMP file.	Sureone	2010-8-3

1. Brief Introduction

The protocol is simplified JetFile II version, and part of the JetFileII full version. It is mainly used to control displaying text file. In the end of the protocol, there are detailed demonstrations to guide for development.

Communication Notes:

For RS232 & RS485 communication: the default communication baud rate is 9600. This baud rate can be changed with software "Sigma" offered by manufacturer (related user's manual for "Sigma" will be offered upon requirement)
For Ethernet communication: Apply TCP/IP communication protocol, 3001 or 9520 port.
Size of communication buffer area: It is suggested to send less than 1024-byte data to the control board every time.

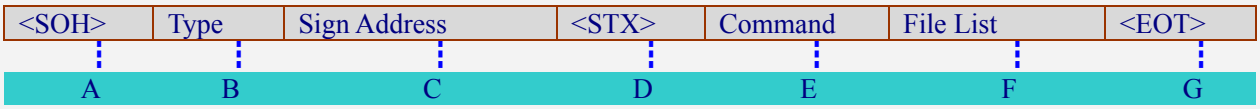
Typeset Notes:

In the document, 0x means hex data, such as <0x30>; ″ ″ or ′′ indicates ASCII characters; others are ten's digits.

2. Command for Play List Control

Play list is used to instruct the sequence of display files. They will be circularly played one by one. If there is no play list, it will circularly play all playable files.

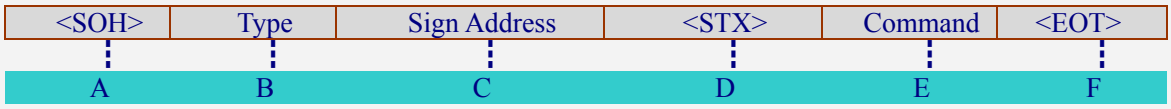
2.1 Format of Writing Play List to Commands



Code	Name	Size	Meaning & Value
A	<SOH>	1 Byte	Beginning communication command, the value is <0x01>
B	Type	1 Byte	Writing type, the value is 'Z'
C	Sign Address	2 Bytes	Unit address of the LED sign, range from 00 to 99. 00 is broadcast address. The value is in ASCII code, for example, 00 addresses are: 0x30,0x30。
D	<STX>	1 Byte	Beginning command, the value is <0x02>
E	Command	4 Byte	Writing play list command, the value is "E.SL"
F	File List	1 | 5 Bytes	Play list It means single Text filename, or 2 filenames with the same path. For example, 02<0x0F>DTAB1 play list means: 1. 0 file in default disk; 2. 2 file in default disk; 3. D:\T\AB file; 4. 1 file in default disk.
G	<EOT>	1 Byte	Ending Command（0x04 | 0x03） 1. Value 0x04 means nothing returns. 2. Value 0x03 means something returns. It will return "OK" if the command is handled successfully.

2.2 Format of Deleting Play List from Commands

Table 2.2.1 Format of Operating Play List Commands



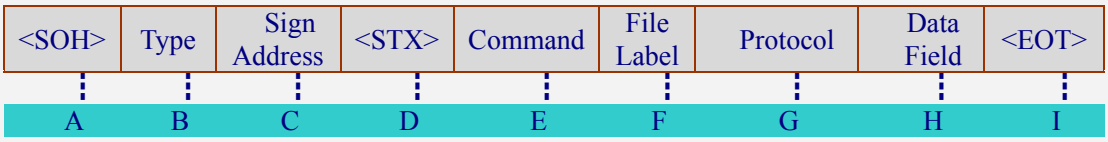
Code	Name	Size	Meaning & Value
A	<SOH>	1 Byte	Beginning Communication Command, the value is <0x01>
B	Type	1 Byte	Writing Type, the value is 'Z'
C	Sign Address	2 Bytes	Unit address of the LED sign, range from 00 to 99. 00 is broadcast address. The value is in ASCII code, for example, 00 addresses are: 0x30,0x30。
D	<STX>	1 Byte	Beginning Command, the value is <0x02>
E	Command	2 Byte	Writing play list command, the value is "E."
F	<EOT>	1 Byte	Ending Command（0x04 | 0x03） 1. Value 0x04 means nothing returns. 2. Value 0x03 means something returns. It will return "OK" if the command is handled successfully.

For example, the command below means deleting play list:
<0x01>Z00<0x02>E.<0x04>

3. Display Control Command

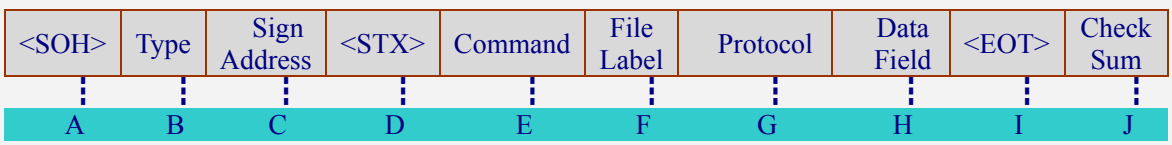
It introduces how to write TEXT files to commands, and the display control commands. The true font of LED sign has to support displaying TEXT file. There are mainly 3 parts: communication format without CheckSum, communication format with CheckSum, as well as display control commands.

3.1. Communication format without CheckSum



Code	Name	Size	Meaning & Value
A	<SOH>	1 Byte	Beginning Communication Command, the value is <0x01>
B	Type	1 Byte	Writing type, the value is 'Z'
C	Sign Address	2 Bytes	Unit address of the LED sign, range from 00 to 99. 00 is broadcast address. The value is in ASCII code, for example, 00 addresses are: 0x30,0x30。
D	<STX>	1 Byte	Beginning Command, the value is <0x02>
E	Command	1 Byte	Writing file command, the value is 'A'
F	File Label	1 | 5 Bytes	File name 1. If the first character is not <0x0F>, the field has only 1 file name. That means writing this file to default disk. 2. If the first character is <0x0F>, it means a path in the back. The format is: <0x0F><disk><contents><file name(as long as 2 characters)> For example, <0x0F>DTT1 means: D---disk D T---contents T T1---file T1 Refer to appendix 1 for valid filename characters.
G	Protocol	1 Bytes	Protocol control command, the value is <0x06>
H	Data Filed	N Byte	Display contents, the value is in ASCII/GB232/BIG5 coding characters. The control command can be inserted (refer to display control command), the size is [1 – 1024] bytes.
I	<EOT>	1 Byte	Ending command（0x04 | 0x03） 1. Value 0x04 means nothing returns. 2. Value 0x03 means something returns. It will return "OK" if the command is handled successfully.

3.2 Communication Format with CheckSum (at present Ver97XX/Ver8EXX/VerA5XX series support it)



Code	Name	Size	Meaning & Value

A	<SOH>	1 Byte	Beginning Communication Command, the value is <0x01>
B	Type	1 Byte	Writing Type, the value is 'Y'
C	Sign Address	2 Bytes	Unit address of the LED sign, range from 00 to 99. 00 is broadcast address. The value is in ASCII code, for example, 00 addresses are: 0x30,0x30。
D	<STX>	1 Byte	Beginning Command, the value is <0x02>
E	Command	1 Byte	Writing file command, the value is 'A'
F	File Label	1|5 Bytes	File name 1. If the first character is not <0x0F>, the field has only 1 file name. That means writing this file to default disk. 2. If the first character is <0x0F>, it means a path in the back. The format is: <0x0F><disk><contents><file name(as long as 2 characters)> For example, <0x0F>DTT1 means: D---disk D T---contents T T1---file T1 Refer to appendix 1 for valid filename characters.
G	Protocol	1 Bytes	Protocol control command, the value is <0x06>
H	Data Filed	N Byte	Display contents, the value is in ASCII/GB232/BIG5 coding characters. The control command can be inserted (refer to display control command), the size is [1 – 1024] bytes.
I	<EOT>	1 Byte	Ending command（0x04 | 0x03） 1. Value 0x04 means nothing returns. 2. Value 0x03 means something returns. It will return "OK" if the command is handled successfully.
J	Checksum	2 Bytes	CheckSum Compute the CheckSum from A to I(including A and I). Low byte is in the front, and high byte is in the back. The algorithm of CheckSum is referred to appendix 2:CheckSum Computing Function.

3.3. Display Control Command

Display control command is to control Text display methods, such as color, fonts etc..

Dec	Hex	description
7	07	**[command for twinking]：2 Bytes format** 0x07 +'0'= ends to flash(default); 0x07 +'1'= begins to flash;
8	08	**[command for line space]：2Bytes format** 0x08 +'0'= Line space is 0； 0x08 +'4'= line space is 4； 0x08 +'7'= line space is 7； 0x08 +'1'= line space is 1； 0x08 +'5'= line space is 5； 0x08 +'8'= line space is 8； 0x08 +'2'= line space is 2； 0x08 +'6'= line space is 6； 0x08 +'9'= line space is 9； 0x08 +'3'= line space is 3；
10	0A	**[command for display modes]：3 Bytes format；'I'=Text Moving in Display Mode, 'O'=Text Moving out Display Mode** + ['I'|'O'] + 0x 2F = random; + ['I'|'O'] + 0x 30 = jump out； + ['I'|'O'] + 0x 31 = move left； + ['I'|'O'] + 0x 32 = move right； + ['I'|'O'] + 0x 33 = scroll left； + ['I'|'O'] + 0x 48 = fan out; + ['I'|'O'] + 0x 49 = fan in; + ['I'|'O'] + 0x 4a = spiral R; + ['I'|'O'] + 0x 4b = spiral L; + ['I'|'O'] + 0x 4c = to four corners; + ['I'|'O'] + 0x 4d = from four corners;

Dec	Hex	description
		+ ['I'|'O'] + 0x 34 = scroll right; + ['I'|'O'] + 0x 35 = move up; + ['I'|'O'] + 0x 36 = move down; + ['I'|'O'] + 0x 37 = scroll to L/R; + ['I'|'O'] + 0x 38 = scroll up; + ['I'|'O'] + 0x 39 = scroll down; + ['I'|'O'] + 0x 3a = scroll from L/R; + ['I'|'O'] + 0x 3b = scroll from U/D; + ['I'|'O'] + 0x 3c = scroll to U/D; + ['I'|'O'] + 0x 3d = shuttle from L/R; + ['I'|'O'] + 0x 3e = shuttle from U/D; + ['I'|'O'] + 0x 3f = peel off L; + ['I'|'O'] + 0x 40 = peel off R; + ['I'|'O'] + 0x 41 = shutter from U/D; + ['I'|'O'] + 0x 42 = shutter from L/R; + ['I'|'O'] + 0x 43 = raindrops; + ['I'|'O'] + 0x 44 = Random Mosaic; + ['I'|'O'] + 0x 45 = twinkling stars; + ['I'|'O'] + 0x 46 = hip-hop; + ['I'|'O'] + 0x 47 = radar scan; + ['I'|'O'] + 0x 4e = to four sides; + ['I'|'O'] + 0x 4f = from four sides; + ['I'|'O'] + 0x 50 = scroll out from four blocks; + ['I'|'O'] + 0x 51 = scroll in to four blocks; + ['I'|'O'] + 0x 52 = move out from four blocks; + ['I'|'O'] + 0x 53 = move in to four blocks; + ['I'|'O'] + 0x 54 = scroll from U/Left, square; + ['I'|'O'] + 0x 55 = scroll from L/Right, square; + ['I'|'O'] + 0x 56 = scroll from U/Left, square; + ['I'|'O'] + 0x 57 = scroll from U/Right, square; + ['I'|'O'] + 0x 58 = scroll from U/Left, Slanting; + ['I'|'O'] + 0x 59 = scroll from U/Right, Slanting; + ['I'|'O'] + 0x 5a = scroll from L/Left, Slanting; + ['I'|'O'] + 0x 5b = scroll from L/Right, slanting; + ['I'|'O'] + 0x 5c = move in from U/Left corner; + ['I'|'O'] + 0x 5d = move in from U/Right corner; + ['I'|'O'] + 0x 5e = move in from L/Left corner; + ['I'|'O'] + 0x 5f = move in from L/Right corner; + ['I'|'O'] + 0x 60 = growing up;
11	0B	**[command for special font]：2 Bytes format** + 0x20 MM/DD/YY + 0x21 DD/MM/YY + 0x22 MM-DD-YY + 0x23 DD-MM-YY + 0x24 MM.DD.YYYY + 0x25 YY + 0x26 YYYY + 0x27 MM (number) + 0x28 MMM.(char) + 0x29 DD (number) + 0x2A day of week (number) + 0x2B day of week (char) + 0x2C HH (24 hours format) + 0x2D MIN + 0x2E SEC + 0x2F HH：MIN + 0x30 HH：MIN (12 AM/PM) +0x31 Temperature(Celsius) +0x32 Humidity +0x33 Temperature(Fahrenheit)
12	0C	**[command fro new frame]：1 Bytes format, Begin a new page**
13	0D	**[command for new line]：1 Bytes format, Begin a new page**
14	0E	**[command for frame pause]：(Different frame can have different pause time)** 0x0E +'0'， time for shot stop(unit: second), range from '00' to '99'(4 Bytes format) 0x 0E +'1'，time for shot stop(unit: millisecond), range from'00'to'99'(4 Bytes format) 0x 0E +'2'，time for long stop(unit: second), range from '0000'-'9999'(6 Bytes format) 0x 0E +'3'，time for shot stop(unit: millisecond), range from '0000'-'9999'(6 Bytes format)

Dec	Hex	description
		For example: Pause for 1 second： 0x0E +‘0’+‘01’ Pause for 12 seconds： 0x0E +‘0’+ ‘12’ Pause for 68 seconds： 0x0E +‘0’+ ‘68’ Pause for 108 seconds： 0x0E +‘2’+ ‘0108’ Pause for 4500 seconds： 0x0E +‘2’+ ‘4500’ Pause for 50 milliseconds： 0x0E +‘1’+ ‘50’ Pause for 88 milliseconds： 0x0E +‘1’+ ‘88’ Pause for 170 milliseconds： 0x0E +‘3’+ ‘0170’
15	0F	**[command for display speed]：2 Bytes format** +‘0’is the fastest speed； +‘1’ is very fast speed； +‘2’ is fast speed； +‘3’is medium speed; +‘4’ slow speed； +‘5’ is very slow speed； +‘6’ is the slowest speed；
20	14	**[command for allocating text and pictures]：** format(3 Bytes format):0x14 + [drive] + [file label] [drive] : the valid values are： ‘D’, ‘E’, ‘F’ [file label]:file name with single character For example, insert a picture file A in drive D: 0x14 + ‘D’ + ‘A’
26	1A	**[command for font and character size]：2 Bytes format** + '0'(0x30) 5 * 5 standard English characters font + '1'(0x31) 7 * 6 standard English characters font + '2'(0x32) 14 * 8 standard English characters font + '3'(0x33) 15 * 9 standard English characters font + '4'(0x34) 16 * 9 standard English characters font + '5'(0x35) 16 * 16 standard Chinese characters font + '6'(0x36) 24 * 16 standard English characters font + '7'(0x37) 24 * 24 standard Chinese characters font + '8'(0x38) 32 * 18 standard English characters font + '9'(0x39) 32 * 32 standard Chinese characters font + ':'(0x3a) 11 * 9 standard English characters font + ';'(0x3b) 12 * 7 standard English characters font + '<'(0x3c) 22 * 18 standard English characters font + '='(0x3d) 30 * 18 standard English characters font + '>'(0x3e) 40 * 21 standard English characters font + 'G'(0x47) 40 SimSun + 'N'(0x4E) 14 * 10 bold English characters font + 'O'(0x4F) 15 * 10 bold English characters font + 'P'(0x50) 16 * 12 bold English characters font + 'Q'(0x51) 24 * 8 bold English characters font + 'R'(0x52) 32 * 8 bold English characters font + 'S'(0x53) 11 * 7 bold English characters font + 'T'(0x54) 12 * 7 bold English characters font + 'U'(0x55) 22 * 12 bold English characters font + 'V'(0x56) 40 * 21 bold English characters font + 'W'(0x57) 24 SimHei + 'X'(0x58) 24 STXinwei + 'Y'(0x59) 24 STXingkai + 'Z'(0x5a) 24 LiSu + '['(0x5b) 24 YouYuan + '\'(0x5c) 32 SimHei + ']'(0x5d) 32 STXinwei + '^'(0x5e) 32 STXingkai + '_'(0x5f) 32 LiSu + '`'(0x60) 32 YouYuan

<table>
<tr><th>Dec</th><th>Hex</th><th>description</th></tr>
<tr><td></td><td></td><td>+ 'H'(0x48) 40 SimHei
+ 'I'(0x49) 40 STXinwei
+ J'(0x4A) 40 STXingkai
+ 'K'(0x4B) 40 LiSu
+ 'L'(0x4C) 40 YouYuan

+ 'a'(0x61) Customer designed character 1
+ 'b'(0x62) Customer designed character 2
+ 'c'(0x63) Customer designed character 3
+ 'd'(0x64) Customer designed character 4
+ 'e'(0x65) Customer designed character 5
+ 'f'(0x66) Customer designed character 6
+ 'g'(0x67) Customer designed character 7
+ 'h'(0x68) Customer designed character 8
+ 'i'(0x69) Customer designed character 9</td></tr>
<tr><td>27</td><td>1B</td><td>[command for setting up type]：3 Bytes format

0x 1B + ‘0’ + ‘a’：No automatic typesetting will be applied. When the display text exceed the display length of the sign, the exceed text would not stop but move left.
0x 1B + ‘0’ + ‘b’：Automatic typesetting will be applied. The default display mode is fixed display.
0x 1B + ‘0’ + ‘c’：reserved
This command should appear before the others.</td></tr>
<tr><td rowspan="2">28</td><td rowspan="2">1C</td><td>[command for color setting]：2

Bytes format

+ 0x30 black
+ 0x31 red color palette
+ 0x32 green color palette
+ 0x33 amber color palette
+ 0x34 mixed colors palette [characters]
+ 0x35 mixed colors palette [horizontal]
+ 0x36 mixed colors palette [wave]
+ 0x37 mixed colors palette [slash]
+ 0x38 blue & white colors palette [vertical gradient]
+ 0x39 blue & white colors palette [horizontal gradient]
+ 0x3A amble & white colors palette [vertical gradient]
+ 0x3B amble & white colors palette [horizontal gradient]
+ 0x3C red & white colors palette [vertical gradient]
+ 0x3D red & white colors palette [horizontal gradient]
+ 0x3E green & white colors palette [vertical gradient]
+ 0x3F green & white colors palette [horizontal gradient]</td></tr>
<tr><td>+ ‘/’ + BGR (24 Bit) user-defined color, B = Blue, G = Green, R = Red, accounts for 1 byte each</td></tr>
<tr><td>29</td><td>1D</td><td>[command for background color setting]:2 Bytes format

+ ‘0’black
+ ‘1’red
+ ‘2’green
+ ‘3’amber
+ ‘/’ + BGR (24 Bit) user-defined color, B=Blue, G=Green, R=Red, accounts for 1 byte each</td></tr>
<tr><td>30</td><td>1E</td><td>[command for align horizontally]:2 Bytes format

+ ‘0’means align centrally(default)
+ ‘1’means align left</td></tr>
</table>

Dec	Hex	description
		+ '2'means align right + '3'reserved [Note]: Different lines can be edited with different align mode. If some lines does not be edited with align mode, these lines would accept the align mode of their previous line as their align mode.
31	1F	**[command for align vertically]:2 Bytes format** + '0'means align centrally(default) + '1'means align left + '2'means align right + '3'reserved
130	82	**[1/2 space]**

4. Command for Writing Simple BMP Pictures

It is used to write BMP pictures to P contents in default or appointed drive. As this command doesn't support sending by separate packages, it can only send some small pictures. The communication formats are as below:

Table 4.1.1 Format of Writing Picture File Command

<SOH>	<REV>	Sign Address	<STX>	Command Code	File Label	Data Field	<EOT>

Name	Size	Start Offset	Meaning
<SOH>	1 Byte	0	Marker bit for beginning command，the <SOH> value is 0x01
<REV>	1 Byte	1	Fixed as'Z'
Sign Address	2 Bytes	2	ＵＵ address of LED sign，range from 00 to 99. 00 is broadcast address. and the value is in ASCII，for example, 00 addresses are 0x30,0x30。
<STX>	1 Byte	4	Beginning command，the <STX> value is 0x02
Command Code	1 Byte	5	Command character, the value is'I'
File Label	1 Bytes	6	File name, as long as only one character. Refer to appendix 1 for valid filenames.
Data Field	N Bytes	7	Picture File data,1024 Bytes at most
<EOT>	1 Byte	7+N	Ending command 1.<0x04> means nothing returns; 2.<0x03> means something returns. It will return "OK" if the command is handled successfully.

Remark for return: <EOT> value 0x04 means nothing returns. <EOT> value 0x03 means something returns. It will return "OK" if the command is handled successfully, or it will return status code. Refer to appendix.

5. Command Demonstration

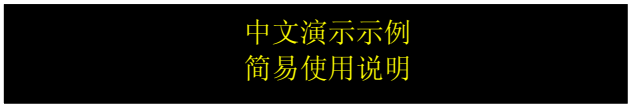
5.1 If you want to display the following text (statically)：

Command:
<0x01>Z00<0x02>AA<0x06><0x0a>I0**ABCD2345<0x0D>EFGH6789<0x0D>IJKL0123**<0x04>
Remark：
When writing contents by broadcast address to file A in default disk of the sign, <0x0a>I0 means the In Mode is Jump Out, and <0x0D> means newline.

5.2 If you want to display the following text (moving left)

Command
<0x01>Z00<0x02>AA<0x06><0x0a>I1**ABCD2345<0x0D>EFGH6789<0x0D>IJKL0123**<0x04>
Remark:
When writing contents by broadcast address to file A in default disk of the sign, <0x0a>I0 means the In Mode is moving left, and <0x0D> means newline.

5.3 Display Two Lines Text



Command:
<0x01>Z01<0x02>AA<0x06><0x08>0<0x0A>I0<0x1C>3 中文演示示例<0x0d>简易使用说明<0x04>
Remark:
When writing contents to file A in default disk of the sign, whose address is 1, <0x0A>I0 means the display mode is jump out, <0x08>0 means line space is 0, <0x0d> means new line, <0x1C>3 means the color is amber.

5.4 Write File BB to RAM Drive

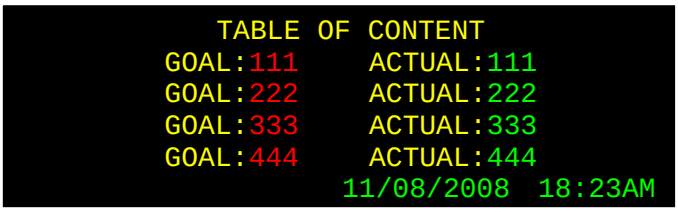
RAM can be erased without limit, however, the file will be deleted after power off. Flash drive has working life limit, but it can save file even power off. Default setting is to write file to flash drive if there is no appointed path. Normally, RAM is E drive, Flash is D drive.

For example, write file BB to RAM drive, and the content is as below:

Command:
<0x01>Z00<0x02>A<0x0f>ETBB<0x06><0x0A>I0<0x1E>1LEFT MSG<0x0d><0x1E>2RIGHT MSG<0x04>
Remark:
When writing contents by broadcast address to file E:\T\BB of the sign, <0x0A>I0 means display mode is jump out, <0x0d> means new line. <0x0f>ETB means file name and its path, refer to format of writing file command. <0x1E>1 and <0x1E>2 means align left and align right respectively.

5.5 Write File CC to Flash Drive

RAM can be erased without limit, however, the file will be deleted after power off. Flash drive has working life limit, but it can save file even power off. Default setting is to write file to flash drive if there is no appointed path. Normally, RAM is E drive, Flash is D drive.
For example, write file CC to Flash drive, and the content is as below:



Command:
<0x01>Z00<0x02>A<0x0f>DTCC<0x06><0x0A>I0<0x1E>0<0x1C>1TABLE OF CONTENT<0x0d>
<0x1C>3GOAL: <0x1C>1111 <0x1C>3ACTUAL: <0x1C>2111<0x0D>
<0x1C>3GOAL: <0x1C>1222 <0x1C>3ACTUAL: <0x1C>2222<0x0D>
<0x1C>3GOAL: <0x1C>1333 <0x1C>3ACTUAL: <0x1C>2333<0x0D>
<0x1C>3GOAL: <0x1C>1444 <0x1C>3ACTUAL: <0x1C>2444<0x0D>
<0x1E>2<0x0B><0x20> <0x0B><0x30><0x04>

Remark:
When writing contents by broadcast address to file D:\T\BB of the sign, <0x0A>I0 means display mode is jump out, <0x0d> means new line. <0x0f>ETB means file name and its path， refer to format of writing file command. <0x1E>1 and <0x1E>2 means align left and align right respectively, <0x1C> means color，<0x0B> means special characters, refer to control commands table.

5.6 Write play list, and, ale the sign to play file A and File B in default drive

Command：
<0x01>Z00<0x02>E.SLAB<0x04>
Remark：
Write play list by broadcast address to the sign, and make the sign to circularly play File A and File B in default drive.

5.7 Write play list, and make the sign to play file A and File D:\T\BB in default drive

Command：
<0x01>Z00<0x02>E.SLA<0x0F>DTBB<0x04>
Remark：
Write play list by broadcast address to the sign, and make the sign to circularly play File A and File D:\T\BB in default drive.

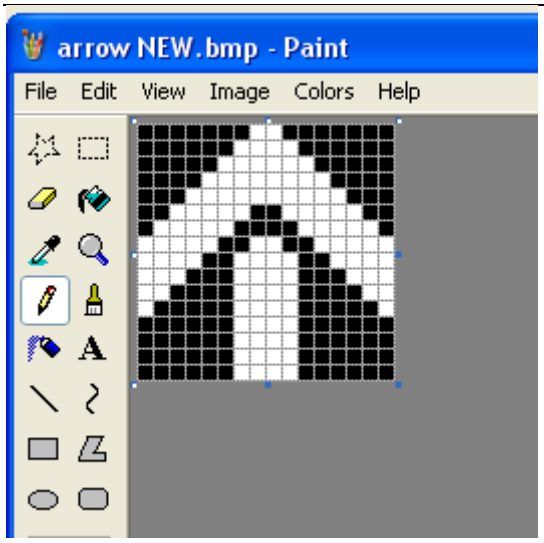
5.8 Write File AA to RAM drive (with CheckSum)

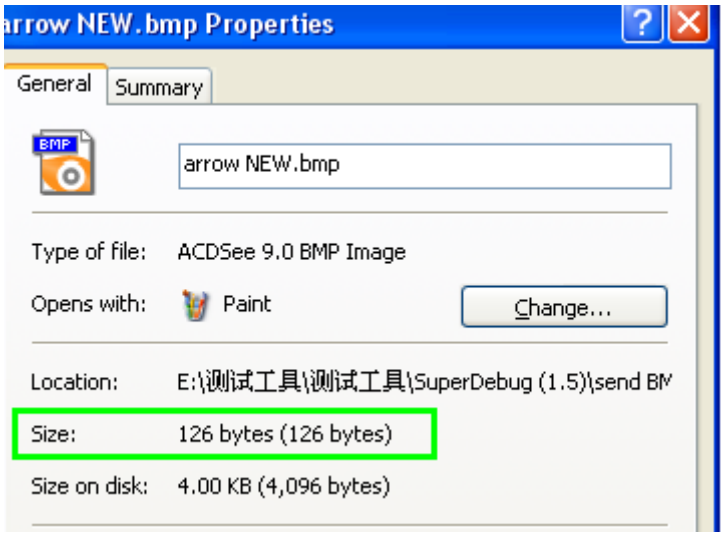
For example, write file AA to RAM drive, and the content is as below:

command：
<0x01>Y01<0x02>A<0x0f>ETAA<0x06><0x0A>I0<0x0A>O0<0x1C>21111<0x03><0x4f><0x04>
Remark：
When writing contents to File E:\T\AA to the No.1 sign, <0x0A>I0<0x0A>O0 means display mode is jump out, <0x0f>ETAA means file and its path, refer to format of writing file command. <0x1C>2 means the color is green, <0x03> means ending, <0x4f><0x04> means CheckSum of 2 bytes, refer to appendix 2 for computing.

5.9 Write a BMP file A to device

5.9.1 Create your BMP file with any editor; please note that, the size of the file should be less than 1024 bytes.

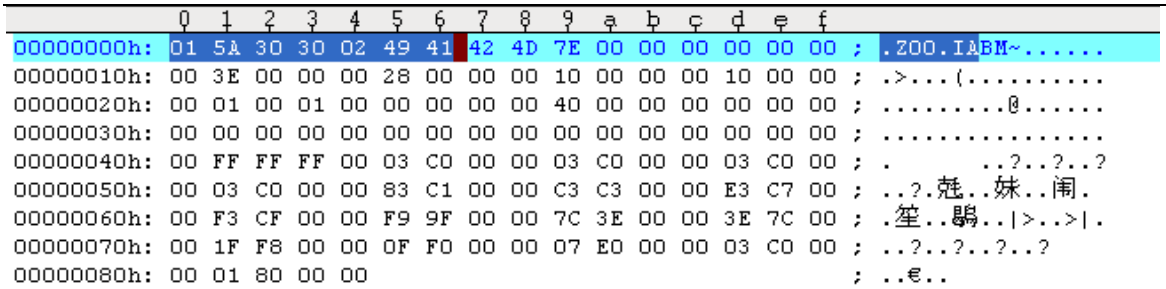




5.9.2 Open it in a Hex editor, like Ultra Edit 32.

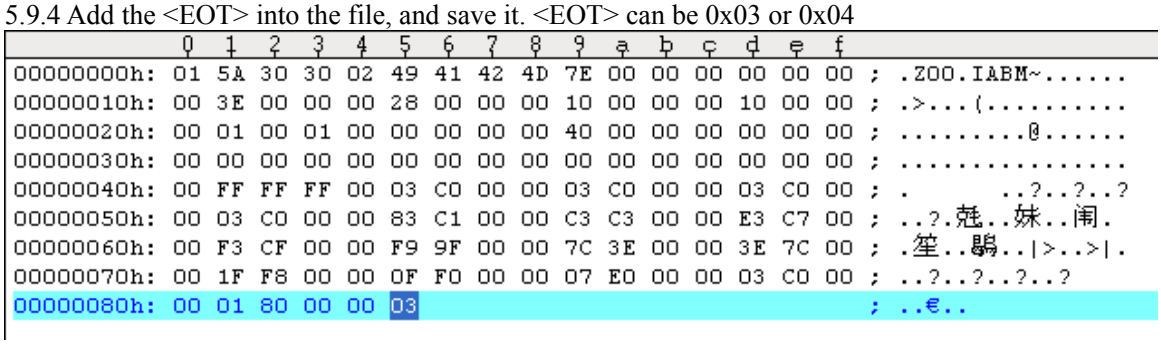
```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 42 4D 7E 00 00 00 00 00 00 00 3E 00 00 00 28 00 ; BM~.......>...(.
00000010h: 00 00 10 00 00 00 10 00 00 00 01 00 01 00 00 00 ; ................
00000020h: 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ..@.............
00000030h: 00 00 00 00 00 00 00 00 00 00 FF FF FF 00 03 C0 ; ..........      ..?
00000040h: 00 00 03 C0 00 00 03 C0 00 00 03 C0 00 00 83 C1 ; ...?..?..?.兎
00000050h: 00 00 C3 C3 00 00 E3 C7 00 00 F3 CF 00 00 F9 9F ; ..妹..闱..笙..鹋
00000060h: 00 00 7C 3E 00 00 3E 7C 00 00 1F F8 00 00 0F F0 ; ..|>..>|...?..?
00000070h: 00 00 07 E0 00 00 03 C0 00 00 01 80 00 00       ; ...?..?..€..
```

5.9.3 Add the <SOH>, <REV>, Sign Address, <STX>, Command Code, and File Label into the file.
For example, name the BMP to be A, and save it into the default device.
These Hex characters will be added:
01 5A 30 30 02 49 41



```
          0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 01 5A 30 30 02 49 41 42 4D 7E 00 00 00 00 00 00 ; .Z00.IABM~......
00000010h: 00 3E 00 00 00 28 00 00 00 10 00 00 00 10 00 00 ; .>...(..........
00000020h: 00 01 00 01 00 00 00 00 00 40 00 00 00 00 00 00 ; .........@......
00000030h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000040h: 00 FF FF FF 00 03 C0 00 00 03 C0 00 00 03 C0 00 ; .       ..?..?..?
00000050h: 00 03 C0 00 00 83 C1 00 00 C3 C3 00 00 E3 C7 00 ; ..?.瓷..妹..闱.
00000060h: 00 F3 CF 00 00 F9 9F 00 00 7C 3E 00 00 3E 7C 00 ; .笙..鹋..|>..>|.
00000070h: 00 1F F8 00 00 0F F0 00 00 07 E0 00 00 03 C0 00 ; ..?..?..?..?
00000080h: 00 01 80 00 00                                  ; ..€..
```

5.9.4 Add the <EOT> into the file, and save it. <EOT> can be 0x03 or 0x04



```
          0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 01 5A 30 30 02 49 41 42 4D 7E 00 00 00 00 00 00 ; .Z00.IABM~......
00000010h: 00 3E 00 00 00 28 00 00 00 10 00 00 00 10 00 00 ; .>...(..........
00000020h: 00 01 00 01 00 00 00 00 00 40 00 00 00 00 00 00 ; .........@......
00000030h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000040h: 00 FF FF FF 00 03 C0 00 00 03 C0 00 00 03 C0 00 ; .       ..?..?..?
00000050h: 00 03 C0 00 00 83 C1 00 00 C3 C3 00 00 E3 C7 00 ; ..?.瓷..妹..闱.
00000060h: 00 F3 CF 00 00 F9 9F 00 00 7C 3E 00 00 3E 7C 00 ; .笙..鹋..|>..>|.
00000070h: 00 1F F8 00 00 0F F0 00 00 07 E0 00 00 03 C0 00 ; ..?..?..?..?
00000080h: 00 01 80 00 00 03                               ; ..€..
```

5.9.5 Send this file to the device.

5.10 Create a message within the BMP file and letters.

5.10.1 Send a BMP file to device; for example, send a BMP file A to the default device.
5.10.2 Create the message, for example, welcome
<0x01>Z00<0x02>AA<0x06>welcome<0x04>
The Hex characters are:
01 5A 30 30 02 41 41 06 77 65 6C 63 6F 6D 65 04

5.10.3 Use the command for allocating text and pictures <0x14> to call the BMP file A.
<0x01>Z00<0x02>AA<0x06><0x14>_Awelcome<0x04>
The Hex characters are:
01 5A 30 30 02 41 41 06 14 5F 41 77 65 6C 63 6F 6D 65 04

Appendix 1 Comparison Table for Valid File Labels and Value

	30H-‘0’	40H-‘@’	50H-‘P’	60H-‘`’	70H-‘p’
21H-‘!’	31H-‘1’	41H-‘A’	51H-‘Q’	61H-‘a’	71H-‘q’
22H-‘“’	32H-‘2’	42H-‘B’	52H-‘R’	62H-‘b’	72H-‘r’
23H-‘#’	33H-‘3’	43H-‘C’	53H-‘S’	63H-‘c’	73H-‘s’
24H-‘$’	34H-‘4’	44H-‘D’	54H-‘T’	64H-‘d’	74H-‘t’
25H-‘%’	35H-‘5’	45H-‘E’	55H-‘U’	65H-‘e’	75H-‘u’
26H-‘&’	36H-‘6’	46H-‘F’	56H-‘V’	66H-‘f’	76H-‘v’
27H-‘‘’	37H-‘7’	47H-‘G’	57H-‘W’	67H-‘g’	77H-‘w’
28H-‘(’	38H-‘8’	48H-‘H’	58H-‘X’	68H-‘h’	78H-‘x’
29H-‘)’	39H-‘9’	49H-‘I’	59H-‘Y’	69H-‘i’	79H-‘y’
2AH-‘*’		4AH-‘J’	5AH-‘Z’	6AH-‘j’	7AH-‘z’
2BH-‘+’	3BH-‘;’	4BH-‘K’	5BH-‘[’	6BH-‘k’	7BH-‘{’
2CH-‘,’	3CH-‘<’	4CH-‘L’		6CH-‘l’	7CH-‘|’
2DH-‘-’	3DH-‘=’	4DH-‘M’	5DH-‘] ’	6DH-‘m’	7DH-‘}’
	3EH-‘>’	4EH-‘N’	5EH-‘^’	6EH-‘n’	
2FH-‘/’	3FH-‘?’	4FH-‘O’	5FH-‘_’	6FH-‘o’	

Appendix 2 CheckSum Computing Function (C language)

The transferring method of CheckSum computing function:
CheckSum = MsgCountCheckSumTwo(buf,0,size – 2) & 0xffff;

```
//*************************************************************
//**** Function Name: MsgCountCheckSumTwo
//**** Function: Compute CheckSum, the return value is 4 bytes, ULONG is 4Byte，UBYTE is 1byte
//**** Parameters:
//****          buf              - Package for checking
//****          begin            - Beginning place
//****          end              - Ending place
//**** return value:
//****          return CheckSum
//*************************************************************
ULONG MsgCountCheckSumTwo(UBYTE *buf, ULONG begin, ULONG end)
{
    ULONG i, check_sum;

    check_sum = 0;
    if (end >= begin)
    {
        i = end - begin;
        buf += begin;
        while(i--)
        {
            check_sum += *buf++;
        }
    }
    return check_sum;
}
```