**2.1 Three Color ESL image format**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Change Description** | **Author** |
| V1.0 | 2018/7/13 | Initial version | Ning |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**C O N F I D E N T I A L**

. **Catalogue**

[1. Purpose 3](#_Toc523664026)

[2. ESL Image Format 3](#_Toc523664027)

# Purpose

This article describes the ESL image encoding format, which is used to guide third parties to develop images that are generated by the tag generated by their own programs.

# ESL Image Format

105, 1

106,2

3

4

5

6

…

104

The display of the 2.1-inch electronic label is displayed from right to left and from top to bottom.

The resolution of the 2.1-inch electronic tag is 212\*104, which is equivalent to 22048 pixels. Each pixel adopts three-color display, corresponding to binary 00(black), 01(white) and red(11). If byte encoding is used, it is equivalent to 1 byte and can represent 4 pixels. That is, it can be represented by 22048/4=5512 bytes.

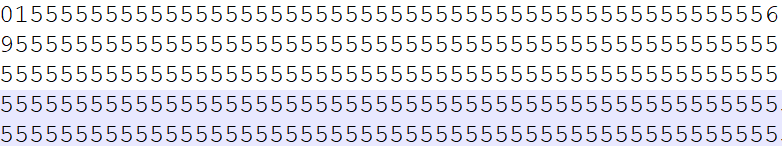
When the cloud initiates an image update to an electronic tag, it needs to send 5512 bytes. Because JSON encoding is used, hexadecimal bytes need to be represented by ASCII, so 1 byte corresponds to 2 ASCII, which is equivalent to a picture corresponding to 5512 \*2 ASCII characters. The following examples illustrate:

Example 1: Need to do the following display

1) The top rightmost column (upper right corner) shows 3 black dots;

2) The bottom right column shows 1 red dot (lower right corner);

3) The top right second column show 1 red dot;



The above one line shows 26 bytes (104 pixels), which corresponds to a column of the 2.1-inch screen.

1) 01: Corresponding binary digit: 0000 0001, that is, the first 3 pixels are 00,00,00, which is black.

2) 56: Corresponding binary digit:

3) 95: Corresponding binary digit:

supplement:

1. We also attach a file "example.bmp" file in the attachment directory, and we also provide the mqtt message that include image content (example.bin.json).

