**4.2 two-color electronic label image encoding format**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Change Description** | **Author** |
| V1.0 | 2018/9/3 | Init version | Ning |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

. **Catalogue**

[1. Purpose 3](#_Toc530924216)

[2. Label 4.2 inch image encoding format 3](#_Toc530924217)

[3. Appendix 4](#_Toc530924218)

# Purpose

This article describes the 4.2inch electronic tag image encoding format, which is used to guide third parties to develop images that use the program to generate tags.

# Label 4.2 inch image encoding format

1, 2, 3, 4, 5, 6, 7…

401, 402, 403

The display of the 4.2-inch electronic label is displayed from left to right and from top to bottom.

The 4.2-inch electronic tag has a resolution of 400\*300, which is equivalent to 120,000 pixels. Each pixel has two-color display, corresponding to binary 0 and 1, where 1 indicates that the pixel is displayed in white and 0 indicates black display. If byte encoding is used, it is equivalent to 1 byte and can represent 8 pixels. That is, it can be represented by 120000/8=15000 bytes.

When the cloud initiates an image update to an electronic tag, it needs to send 15000 bytes. Since JSON encoding is used, hexadecimal bytes are required in ASCII, so 1 byte corresponds to 2 ASCII, which is equivalent to 15000\*2 ASCII characters for one picture. So we suggestion using compression coding, which can greatly reduce the number of characters.

The following examples illustrate:

Example 1: If we want do the following display



1) Display 1 black dot on the far left (upper left corner) of the 1st line, then 1 white point, then 1 black dot;

2) Display 1 black dot on the far right (upper right corner) of the 1st line.

3) Display 1 black dot on the far left of the 2nd line.

The corresponding code is as follows:



For 4.2inch two color ESL, one row need 50 bytes code(400 pixels)

The First line:

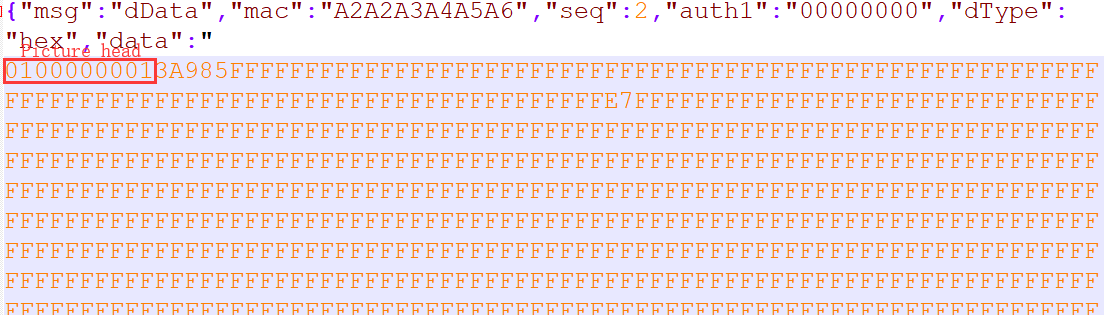
* First byte was 0x5f: Corresponding binary digits: 0101 1111, it means the first pixel was black(0), and the third pixel was black.
* Last byte was 0xFE, Corresponding binary digits: 1111 1110, it means the last pixel was black(0).

The Second line:

* First byte was 0x7f: Corresponding binary digits:

# Appendix

1. We also attach a file "4.2inchExample.bmp" file in the attachment directory, and we also provide the mqtt message that include image content (4.2inchExample.bmp.json).



2. We also attach a file “4.2inchExample.bmpz.json” in the same directory. This file was compression encoding. You can see that the file size is very small. We recommend using compressing encoding json message for 4.2inch ESL when sending picture to ESL.