**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](#_Toc523732484)

[2. Label 4.2 inch image encoding format 3](#_Toc523732485)

# 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

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:





The above line shows 16 bytes (128 pixels).

1) 0x5f: Corresponding binary digits: 0101 1111, which is the first one, and the third pixel is black.

2) The address is 0x32 (50\*8=400). It is the last pixel in the first line. The value is 0xfe, and the corresponding bit position is 11111110, that is, the last pixel bit is black.

3) 0x7f: Corresponding binary digit: 0111 1111, that is, the first pixel of the second row is black.

All the pixels next are displayed in white, which is 0xff.

supplement:

1. There is a file "4.2inchExample.bmp" file in the attachment directory, and the converted bin file (4.2inchExample.bin). The bin file can be viewed with the UltraEdit tool and the image content in the mqtt message (4.2inchExample.bin) .json)

2. You can use the Image2Lcd tool to convert 4.2inchExample.bmp to a bin file, and then download UltraEdit (you can edit the hex tool) to view it.