

LED Control

Tony Chen, Leo Wang

September 27, 2021

Abstract

This document gives an easy guideline on how to send request to LED with panel and code.

1 Software List

- LED Test tool (in LED package)

2 Panel GuideLine

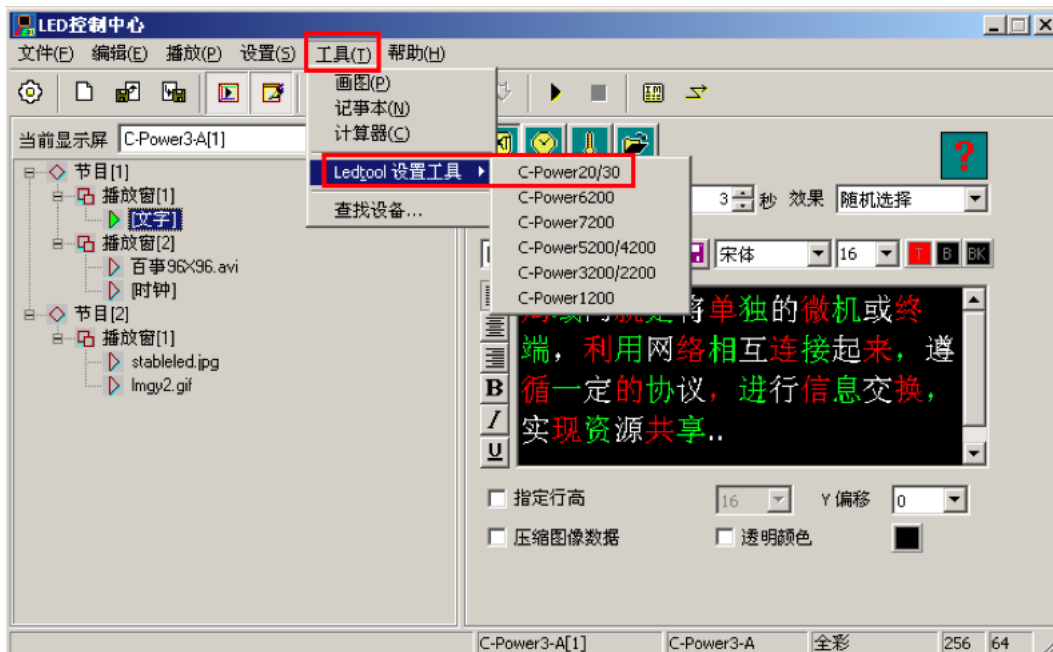


Figure 1: Main Page

- Step 1. Open LED tool, you will see the Main Page like the image

- Step 2. Click on LED Tool and check the version

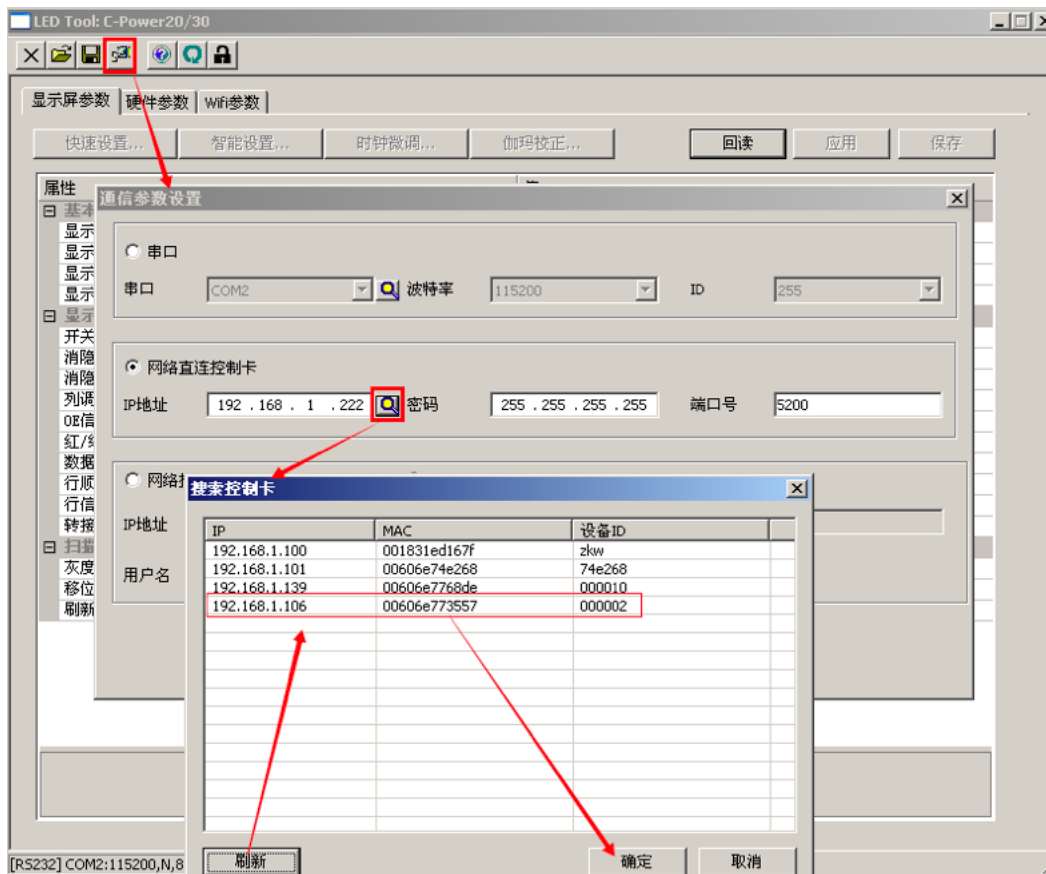


Figure 2: Ip setting of the LED

Usage: Config the LED setting

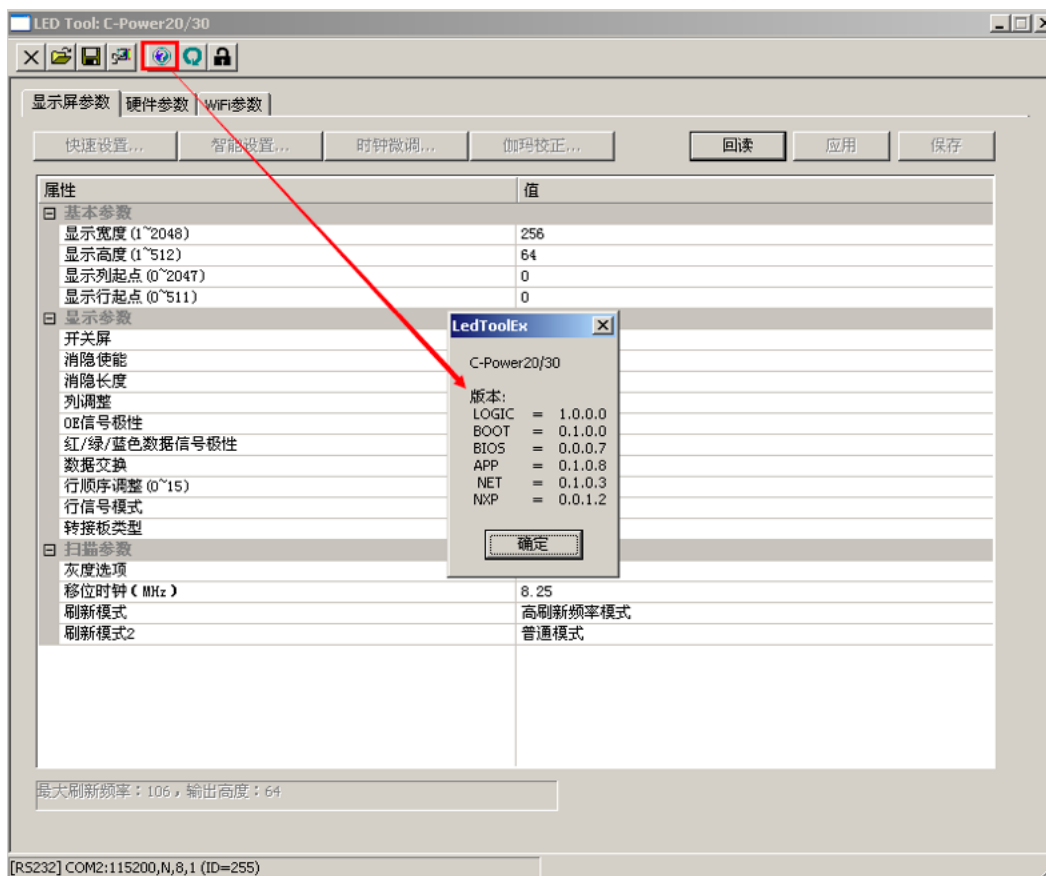


Figure 3: Check the version of the panel

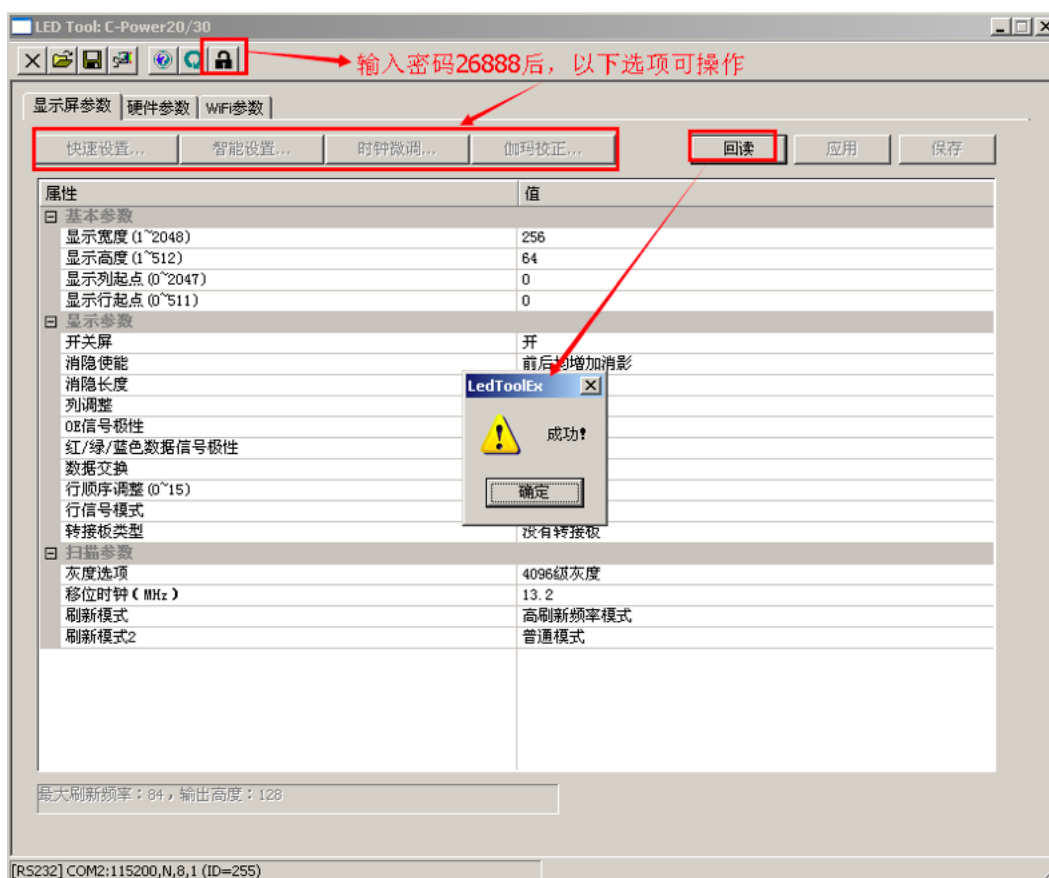


Figure 4: Config of the LED

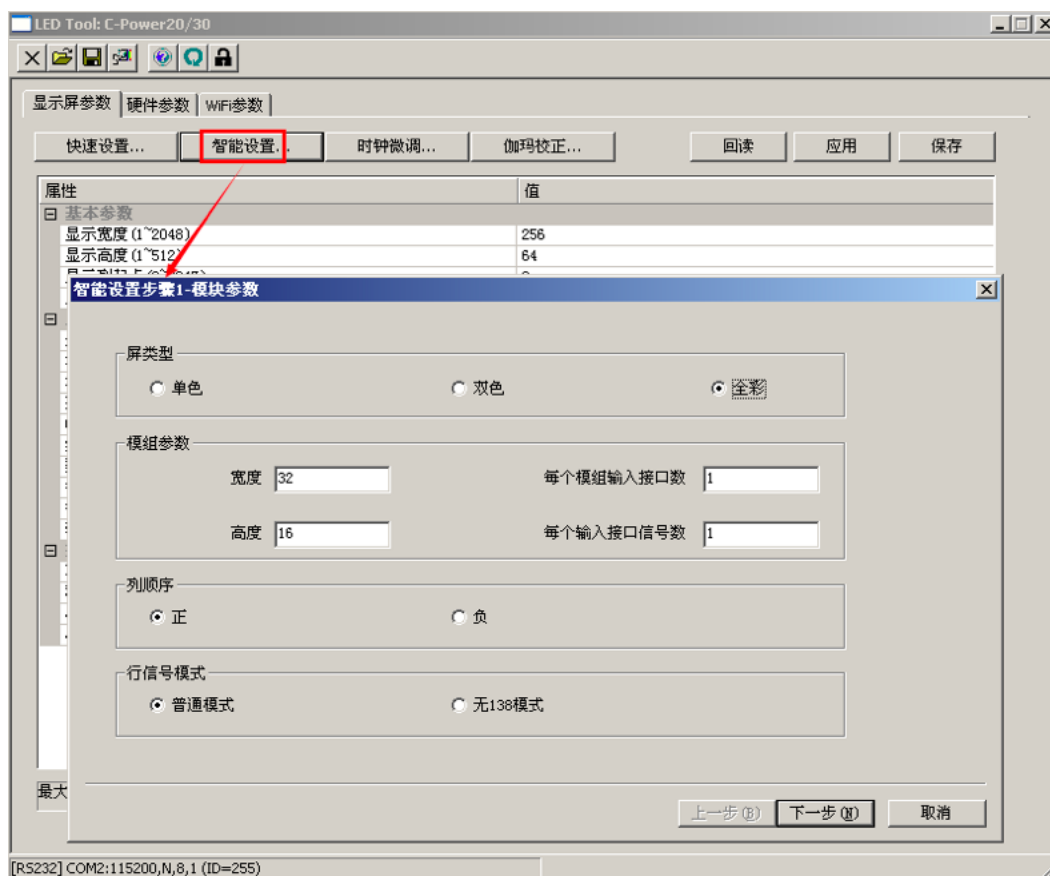


Figure 5: Config of the LED

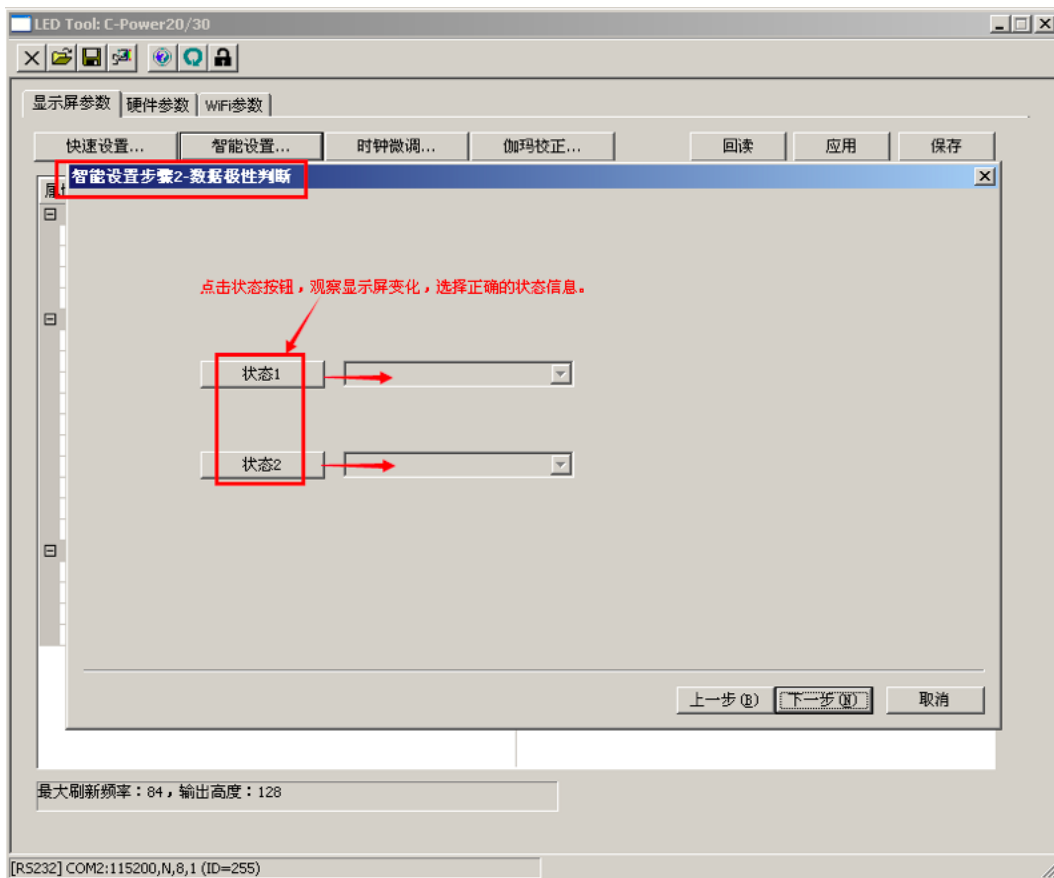


Figure 6: Config of the LED

You could set different condition by configuration

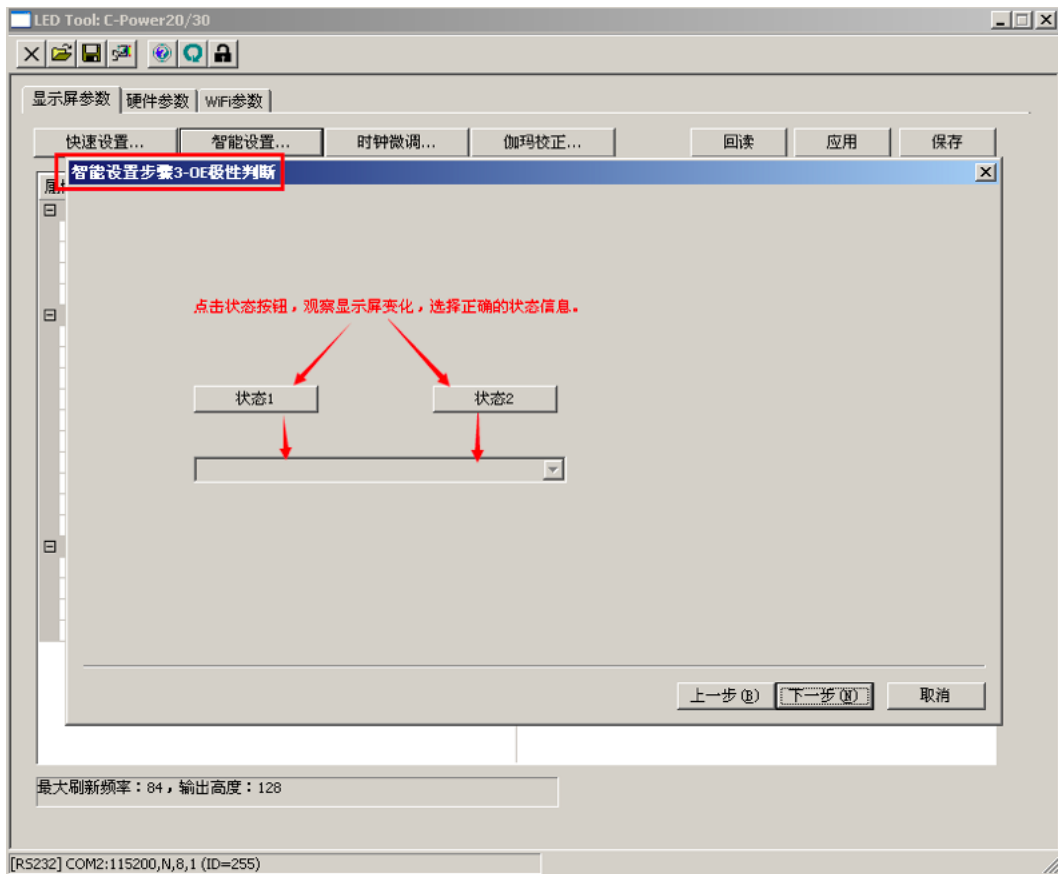


Figure 7: Config of the LED

You could set different condition by configuration

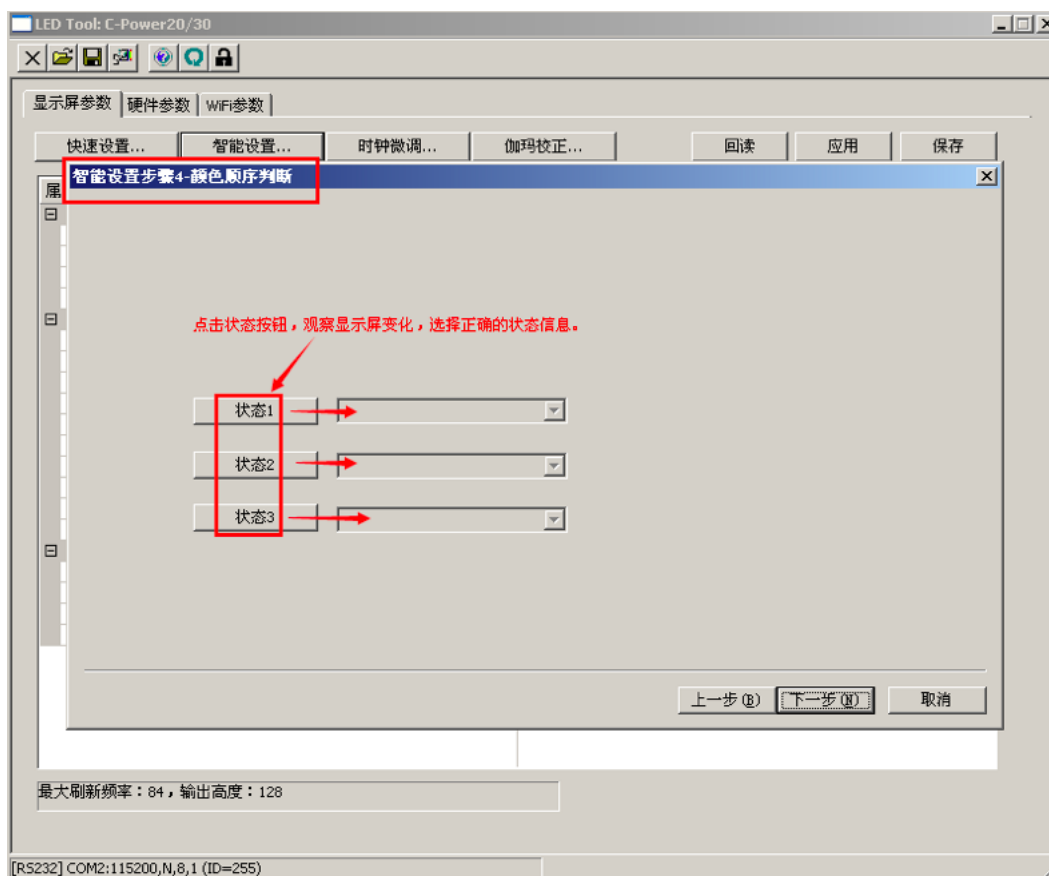


Figure 8: Config of the LED

This will makes the difference on the color

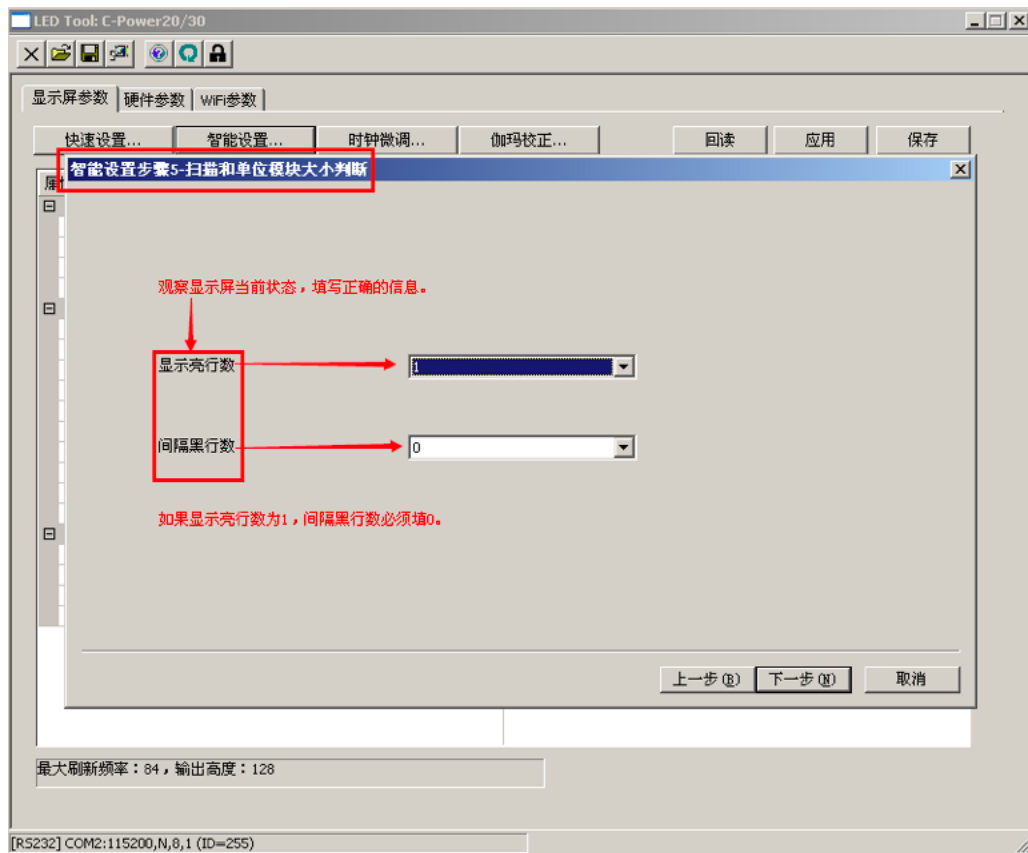


Figure 9: Config of the LED

This will makes the difference on the Line

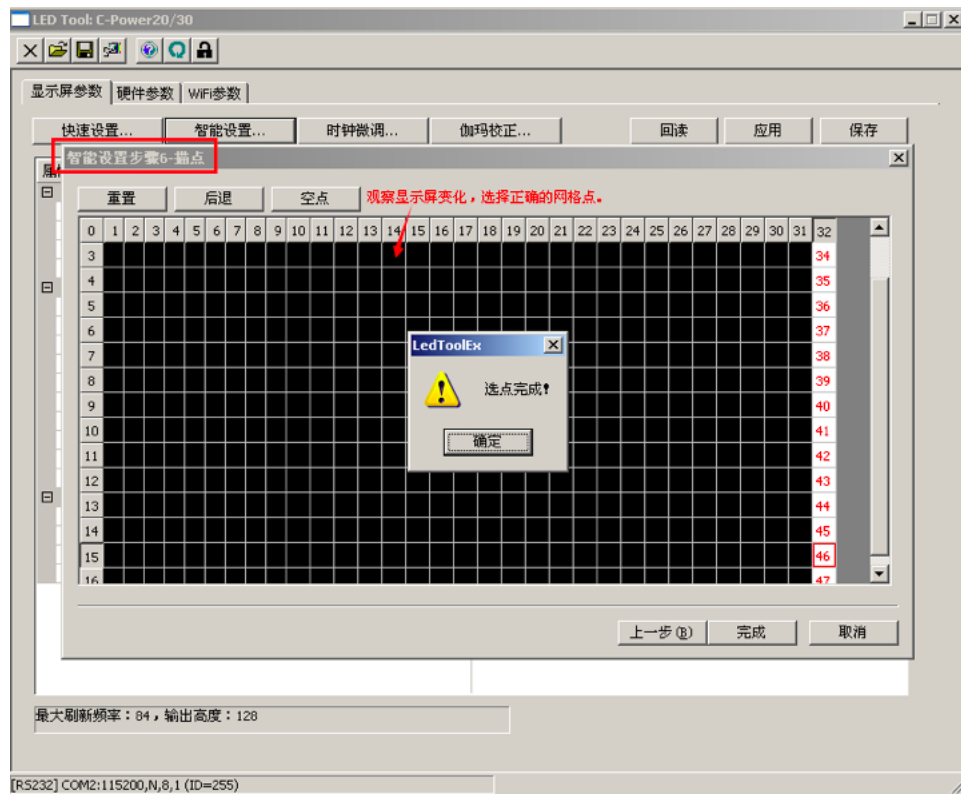


Figure 10: Config of the LED

This will change the different point

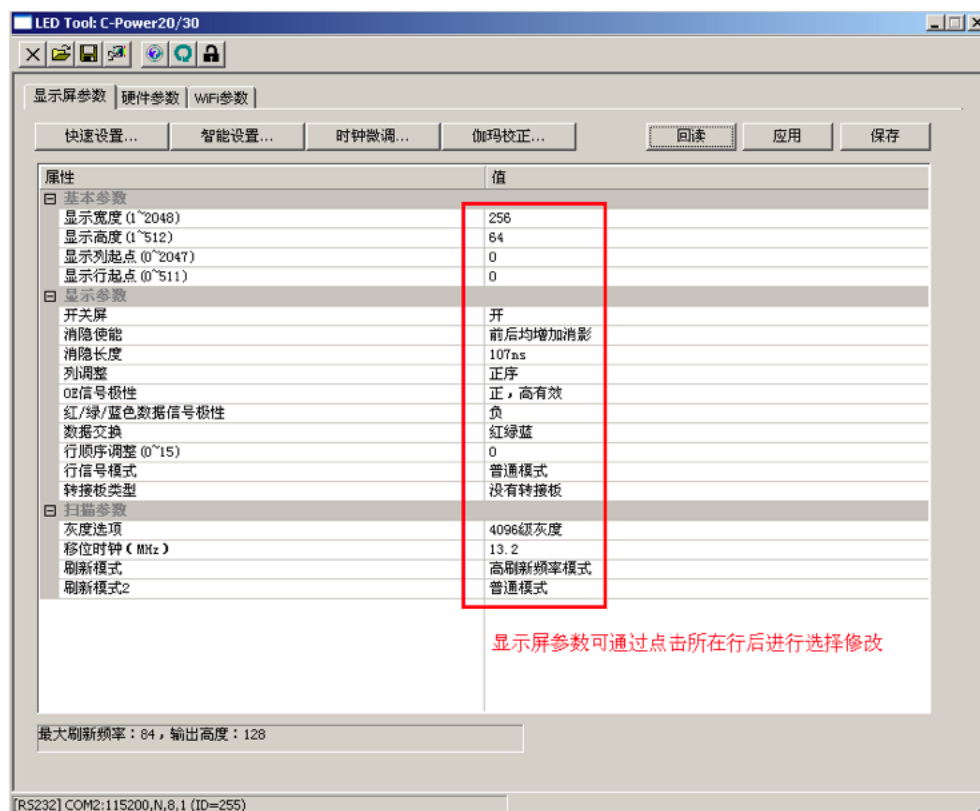


Figure 11: Config of the LED

This will change the different config

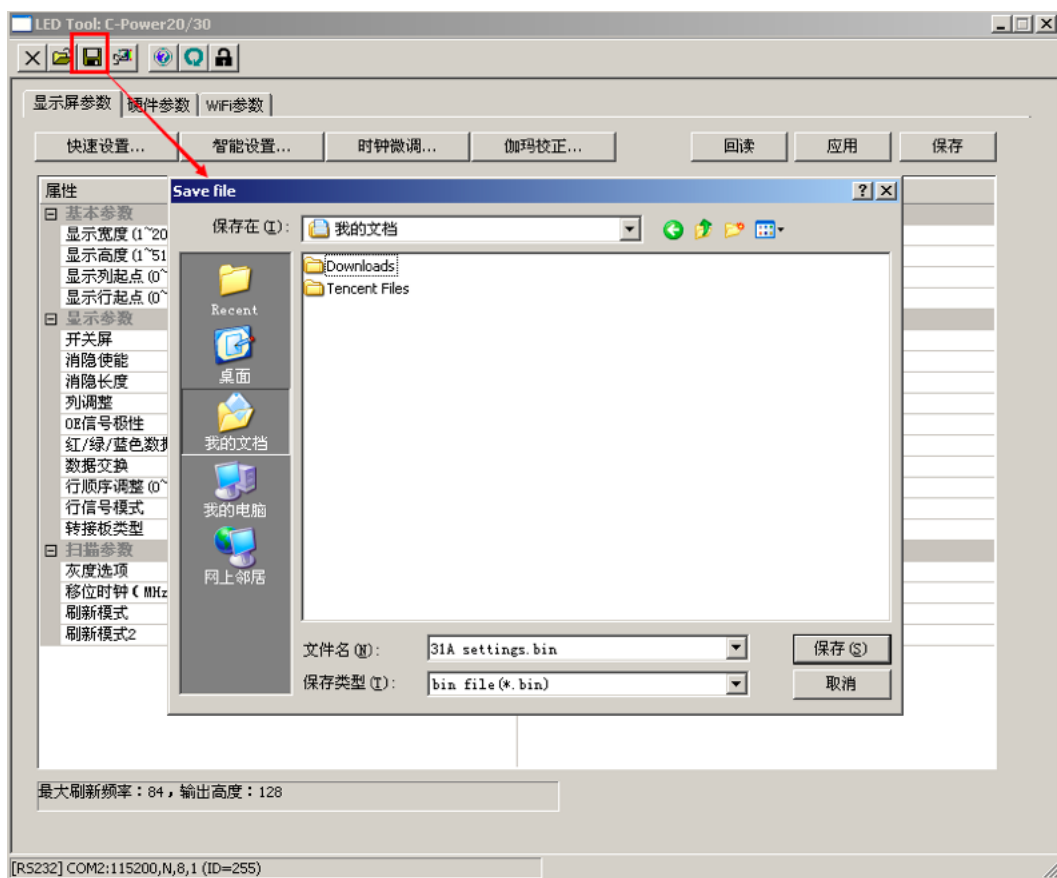


Figure 12: Save the information

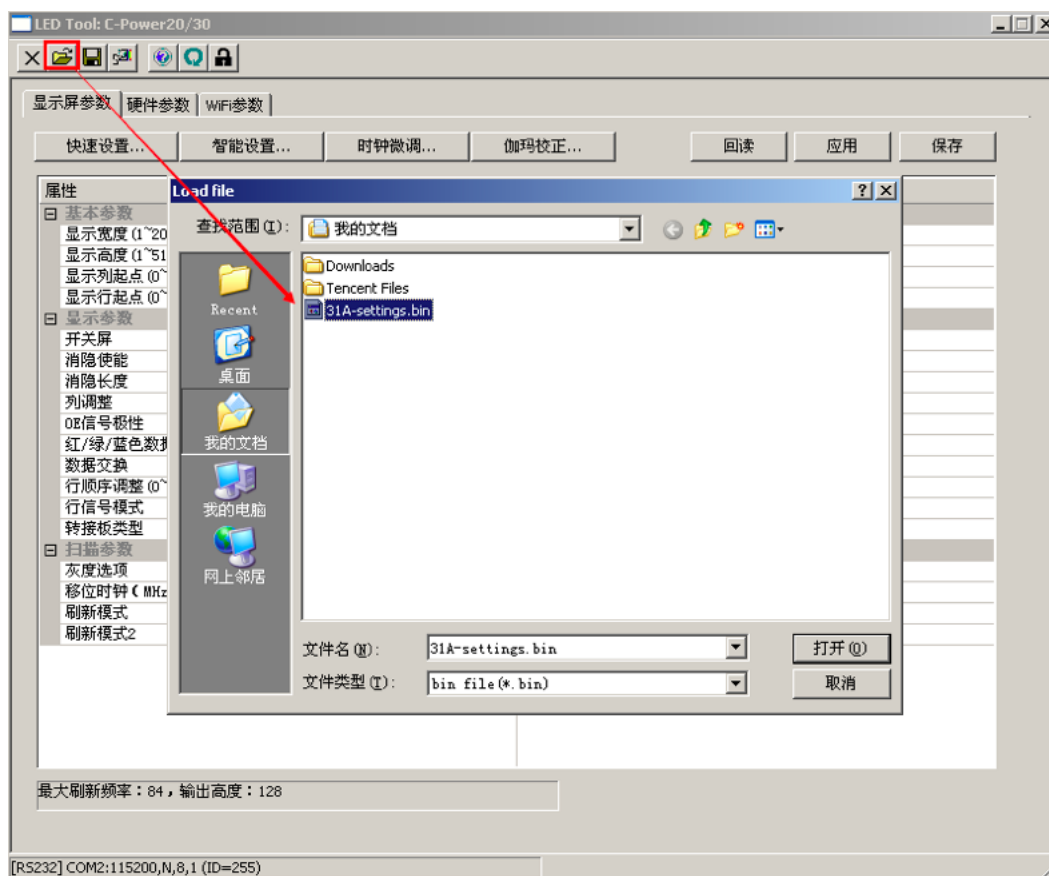


Figure 13: Save the information

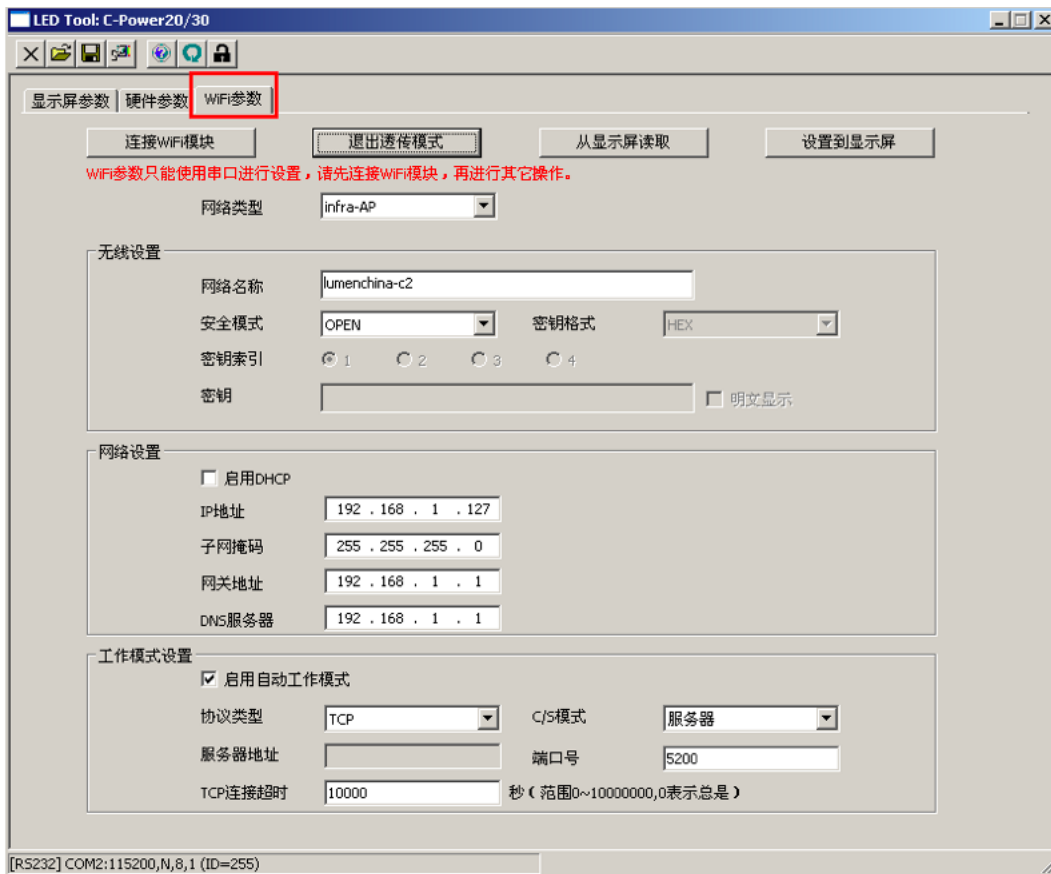


Figure 15: wifi setting

3 Software Approach

```
main.h
1  #pragma once
2
3  int config_led_count;
4  struct config_led_type config_led[MAX_LEDS];
5  SOCKET led_sockfd[MAX_LEDS];
6  SOCKADDR_IN send_to_led[MAX_LEDS];
7
8  char arrow_filename[][100] = { "/opt/led/ArrowLeft.gif" , "/opt/led/ArrowUp.gif" , "/opt/led/ArrowRight.gif" , "/opt/led/ArrowDown.gif" }
```

Figure 16: Main H

```
main.cpp
1  #include "pparking.h"
2  #include "main.h"
3
4  void read_config_led()
5  {
6      int n;
7      char tmpstr[500];
8      char valstr1[500];
9      char valstr2[500];
10     FILE *fp;
11
12     fp = fopen("/lpr_data/config/config_led.ini", "r");
13     if (fp == NULL) {
14         printf("config_led.ini not exist\n");
15         exit(-1);
16     }
17     fscanf(fp, "%s %s\n", tmpstr, valstr1); config_led_count = atoi(valstr1);
18     for (int i = 0; i < config_led_count; i++) {
19         fscanf(fp, "%s %s %s %s %s\n", tmpstr, config_led[i].name, config_led[i].ip, valstr1, valstr2);
20         config_led[i].floor_id = atoi(valstr1);
21         config_led[i].arrow = atoi(valstr2);
22     }
23     fclose(fp);
24 }
25
26 int main()
27 {
28     read_config_led();
29     for (int i = 0; i < config_led_count; i++) {
30         init_led(i);
31         show_led(i, 999);
32     }
33 }
```

Figure 17: Main CPP