DESCRIPTION

IS31FL3194 is a 3-channel LED driver which features two-dimensional auto breathing mode. It has Pattern Mode and Current Level Mode for RGB lighting effects. The maximum output current can be adjusted in 4 levels (40mA Max.).

In Current Level Mode, the current level of each output can be independently programmed and controlled in 256 steps to simplify color mixing. In Pattern Mode, the timing characteristics for output current - current rising (T1), holding (T2), falling (T3) and off time (TS, TP, T4), can be adjusted individually so that each output can independently maintain a pre-established pattern achieving mixing color breathing or a single color breathing without requiring any additional interface activity, thus saving valuable system resources.

FEATURES

- 2.7V to 5.5V supply voltage
- One group RGB/RG+W, or 3 single color LED breathing system-free pre-established pattern
- I2C interface, automatic address increment function
- 4 band programmable output current for each output, each band has 256 current levels
- Selectable gamma value for automatic breathing for each output
- Each pattern have 3 pre-established color

QUICK START

Figure 1: Photo of IS31FL3194 Evaluation Board

RECOMMENDED EQUIPMENT

- 5.0V, 1A power supply

ABSOLUTE MAXIMUM RATINGS

- ≤ 5.5V Micro USB DC power supply

Caution: Do not exceed the conditions listed above, otherwise the board will be damaged.

PROCEDURE

The IS31FL3194 evaluation board is fully assembled and tested. Follow the steps listed below to verify board operation.

Caution: Do not turn on the power supply until all connections are completed.

1) Short last two pins (Bottom & Left) of TP1 to enable the control of board MCU (default status).
2) Connect the 5VDC power to VCC/GND of TP1, or plug in the USB power input to micro-USB.
3) Turn on the power supply, pay attention to the supply current. If the current exceeds 1A, please check for circuit fault.

EVALUATION BOARD OPERATION

The IS31FL3194 evaluation board has five display modes. Press K1 to switch configurations:

Note: See Appendix for each mode's detail.

1) 3 lamps breath one by one
2) Single lamp breath and all lighting
3) RGB breath on high speed
4) RGB breath on medium speed
5) RGB breath on low speed

Note: IS31FL3194 solely controls the FxLED function on the evaluation board.

ORDERING INFORMATION

Part No.	Temperature Range	Package
IS31FL3194-CLS2-EB	-40°C to +85°C, Industrial	WCSP-8, Lead-free

Table 1: Ordering Information

For pricing, delivery, and ordering information, please contacts ISSI's analog marketing team at analog@issi.com or (408) 969-6600.

SOFTWARE CONTROL

Last two pins of TP1 default setting is closed (short). If it is set to open, the MCU's SDB, SCL and SDA pin will be high impedance (open-drain) and external control is allowed.

Follow the steps listed below for external control.

1) Open last two pins of TP1 to enable external control.
2) Pull-up the SDB to VCC or external IO control (H for normal operation).
3) Connect the 5VDC power to the connector.
4) Turn on the power supply/Plug in the Micro USB Pay attention to the supply current. If the current exceeds 1A, please check for circuit fault.
5) Start external IIC control.

Caution: If last two pins of TP1 is closed (shorted), user can't connect the user's MCU, otherwise the user's MCU (maybe 1.8V) will connect to evaluation bard's MCU (3.0V) and maybe damaged.

Please refer to the datasheet to get more information about IS31FL3194.

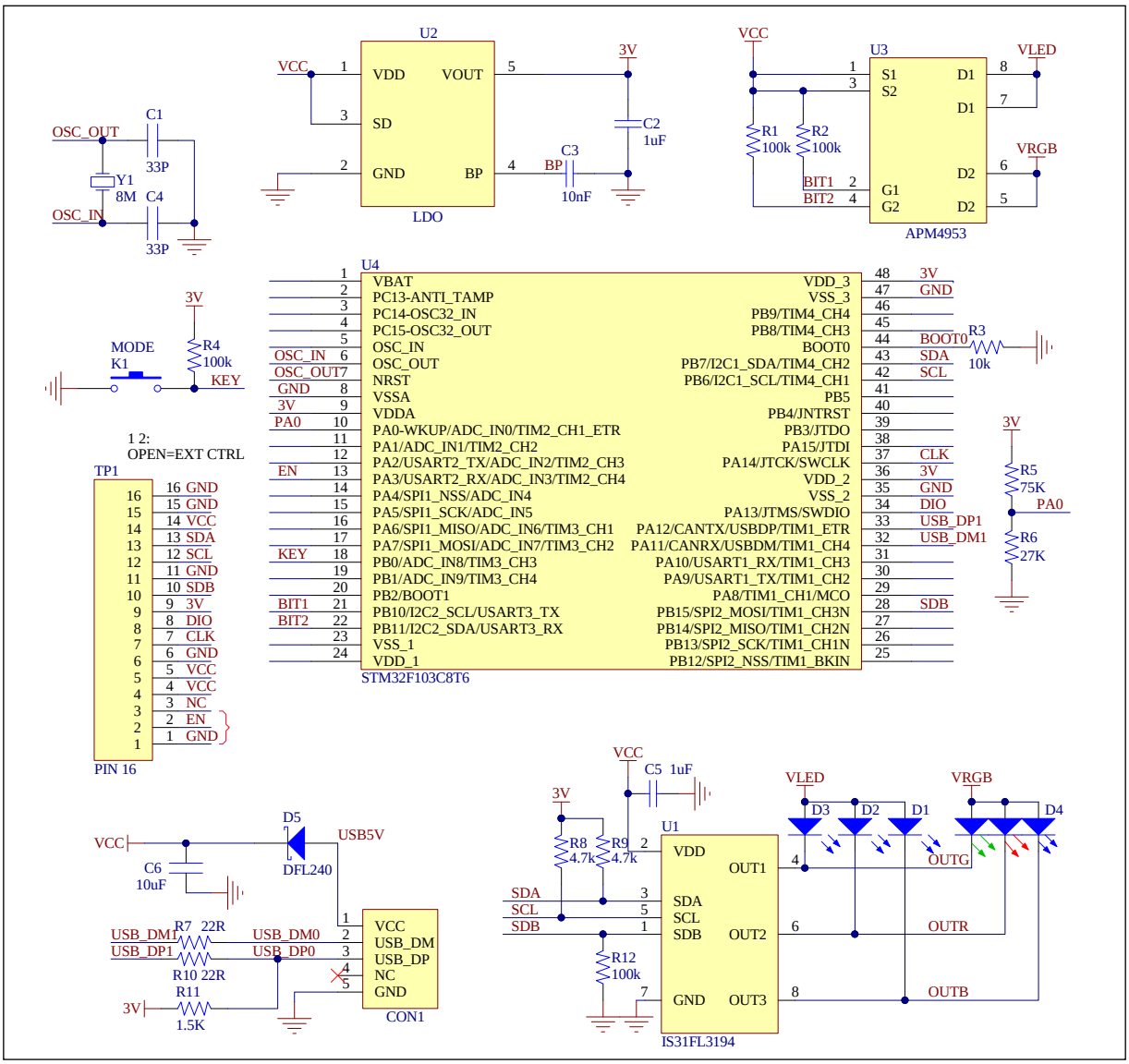


Figure 2: IS32FL3194 Application Schematic

BILL OF MATERIALS

Name	Symbol	Description	Qty	Supplier	Part No.
LED Driver	U1	Matrix LED Driver	1	ISSI	IS31FL3194
LDO	U2	3.0V LDO	1	SGMICRO	SGM2019-3.0YN5G
PMOS	U3	PMOS	1	ANPEC	APM4953
MCU	U4	Microcontroller	1	STM	STM32F103C8T6
LED	D1,D2,D3	LED, SMD Blue	3	EVERLIGHT	19-217/BHC-AN1P2/3T
RGB LED	D4	RGB LED, SMD	1	ROHM	SMLV56RGB1W1
Diode	D5	Diode, SMD	1	DIODES	DFLS240
Crystal	Y1	Crystal, 8MHz	1	HLX	HC-49S
Resistor	R1,R2,R4,R12	RES,100k,1/16W,±5%,SMD	4	Yageo	RC0603JR-07100KL
Resistor	R3	RES,10k,1/16W,±5%,SMD	1	Yageo	RC0603JR-0710KL
Resistor	R5	RES,75k,1/16W,±5%,SMD	1	Yageo	RC0603JR-0775KL
Resistor	R6	RES,27k,1/16W,±5%,SMD	1	Yageo	RC0603JR-0727KL
Resistor	R7,R10	RES,22R,1/16W,±5%,SMD	2	Yageo	RC0603JR-0722RL
Resistor	R8,R9	RES,4.7K,1/16W,±5%,SMD	2	Yageo	RC0603JR-074K7L
Resistor	R11	RES,1.5K,1/16W,±5%,SMD	1	Yageo	RC0603JR-071K5L
Capacitor	C1,C4	CAP,33pF,16V,±20%,SMD	2	Yageo	CC0603KKX7R9BB330
Capacitor	C2,C5	CAP,1μF,16V,±20%,SMD	2	Yageo	CC0603KKX7R9BB105
Capacitor	C3	CAP,10nF,16V,±20%,SMD	1	Yageo	CC0603KKX7R9BB103
Capacitor	C6	CAP,10μF,16V,±20%,SMD	1	Yageo	CC0805KKX7R9BB106
Button	K1(Bottom)	Button	1		

Bill of Materials, refer to Figure 1 above.

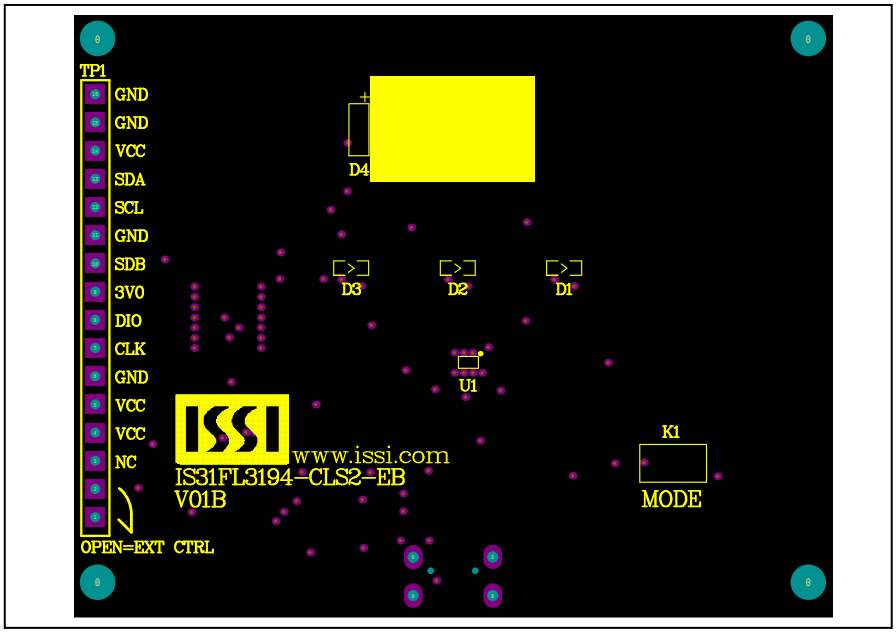


Figure 3: Board Component Placement Guide - Top Layer

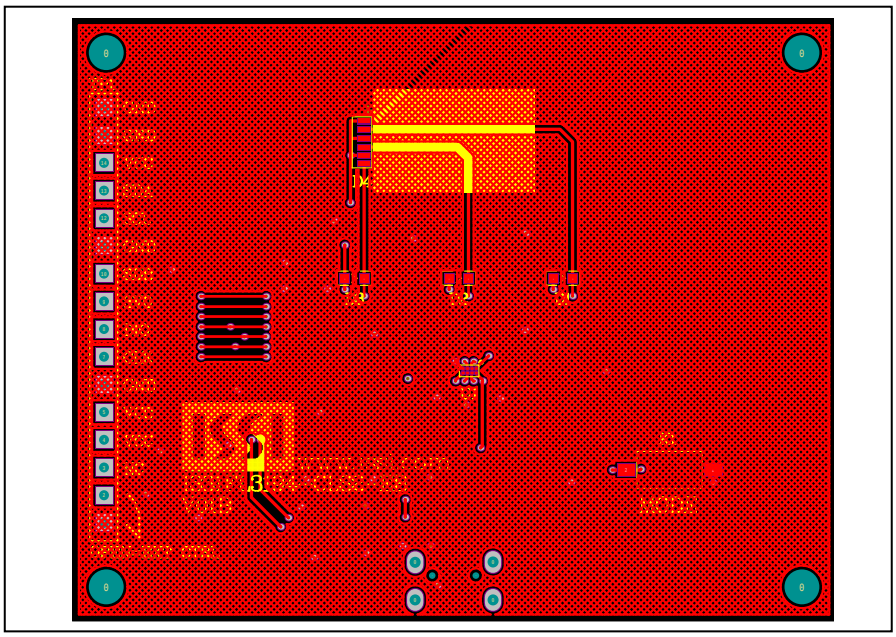


Figure 4: Board PCB Layout - Top Layer

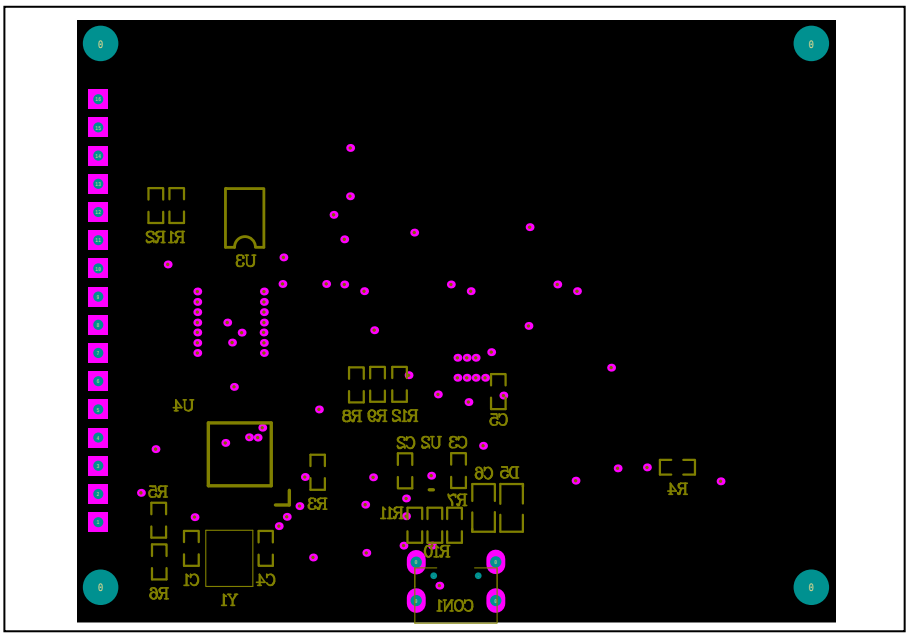


Figure 5: Board Component Placement Guide - Bottom Layer

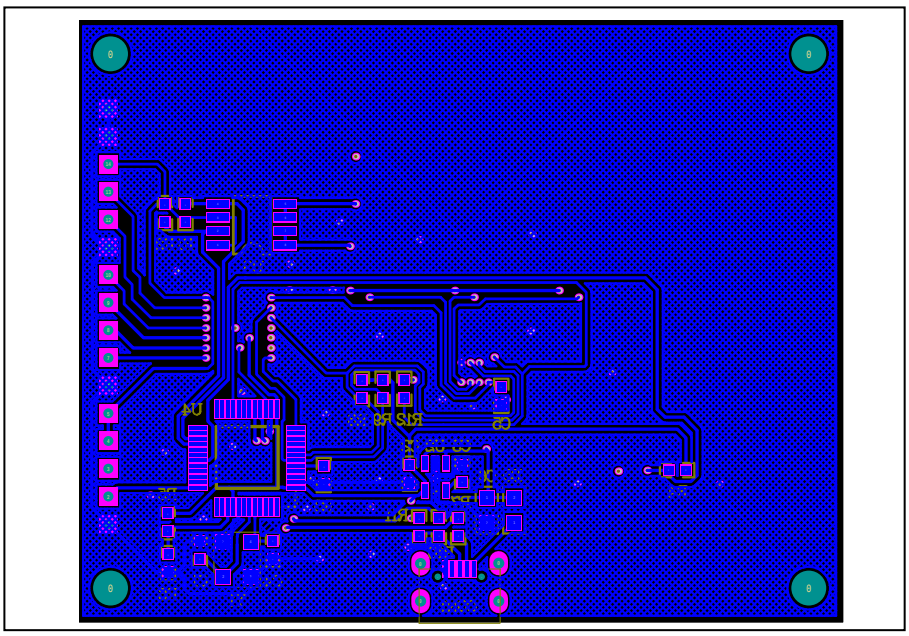


Figure 6: Board PCB Layout - Bottom Layer

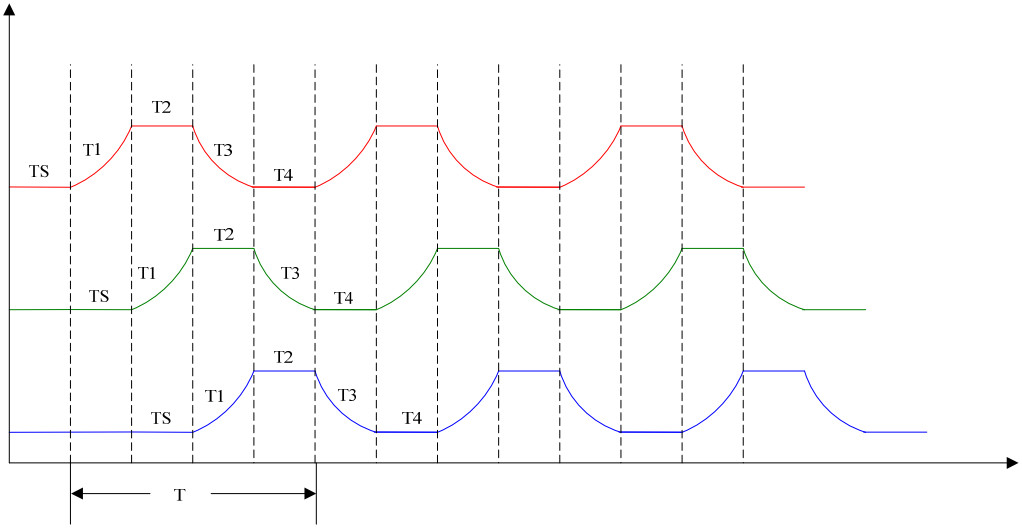
REVISION HISTORY

Revision	Detail Information	Date
A	Initial release	2017.03.23
B	Add appendix	2017.08.09

Appendix:

MODE 1

TS = T1 = T2 = T3 = T4 = 0.51S

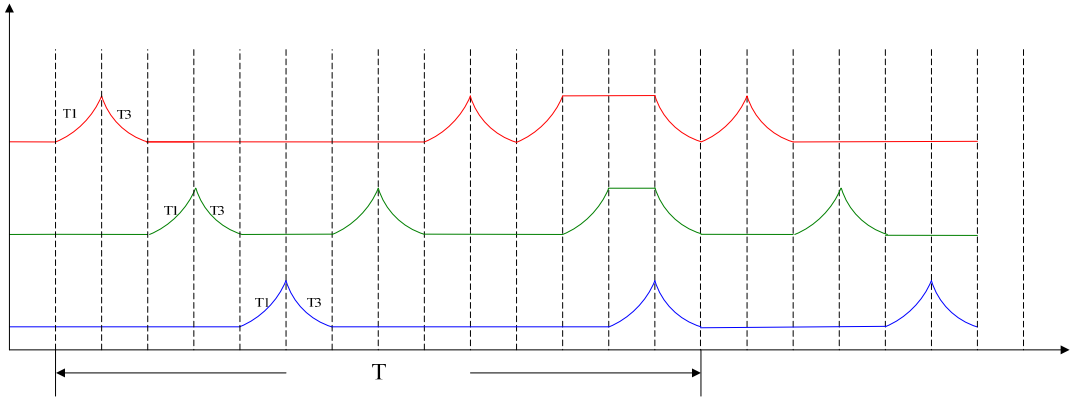



```
//Init
   I2C_WriteByte(Addr_VCC_3194, 0x01, 0x71);//current single mode, normal operation
   I2C_WriteByte(Addr_VCC_3194, 0x02, 0x07);//channel enable
   I2C_WriteByte(Addr_VCC_3194, 0x03, 0x3F);//chx max current 10mA 10mA 10mA
   I2C_WriteByte(Addr_VCC_3194, 0x04, 0x00);// Hold function disable
//pattern 1color
   I2C_WriteByte(Addr_VCC_3194, 0x10, 0x7f);// color 1
   I2C_WriteByte(Addr_VCC_3194,0x11,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x12,0x7f);
   I2C_WriteByte(Addr_VCC_3194, 0x13, 0x7f);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x14,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x15,0x7f);
   I2C_WriteByte(Addr_VCC_3194, 0x16, 0x7f);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x17,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x18,0x7f);
//pattern 2 color
   I2C_WriteByte(Addr_VCC_3194, 0x20, 0x7f);// color 1
   I2C_WriteByte(Addr_VCC_3194,0x21,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x22,0x7f);
   I2C_WriteByte(Addr_VCC_3194, 0x23, 0x7f);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x24,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x25,0x7f);
   I2C_WriteByte(Addr_VCC_3194, 0x26, 0x7f);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x27,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x28,0x7f);
//pattern 3 color
   I2C_WriteByte(Addr_VCC_3194, 0x30, 0x7f);// color 1
   I2C_WriteByte(Addr_VCC_3194, 0x31, 0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x32,0x7f);
   I2C_WriteByte(Addr_VCC_3194, 0x33, 0x7f);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x34,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x35,0x7f);
   I2C_WriteByte(Addr_VCC_3194, 0x36, 0x7f);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x37,0x7f);
   I2C_WriteByte(Addr_VCC_3194,0x38,0x7f);
//Pattern 1 timing
   I2C_WriteByte(Addr_VCC_3194,0x19,0x44);//T1&Ts = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x1A,0x44);//T2&T3 = 0.51S
```

```
   I2C_WriteByte(Addr_VCC_3194,0x1B,0x44);//T4&TP = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x1D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x1E,0x00);//Gamma=2.4
    I2C_WriteByte(Addr_VCC_3194,0x1F,0x00);//Endless time
// Pattern 2 timing
   I2C_WriteByte(Addr_VCC_3194,0x29,0x44);//T1&Ts = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x2A,0x44);//T2&T3 = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x2B,0x44);//T4&TP = 0.51S
    I2C_WriteByte(Addr_VCC_3194,0x2D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x2E,0x00);//Gamma=2.4
    I2C_WriteByte(Addr_VCC_3194,0x2F,0x00);//Endless time
// Pattern 3 timing
   I2C_WriteByte(Addr_VCC_3194,0x39,0x44);//T1&Ts = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x3A,0x44);//T2&T3 = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x3B,0x44);//T4&TP = 0.51S
   I2C_WriteByte(Addr_VCC_3194,0x3D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x3E,0x00);//Gamma=2.4
    I2C_WriteByte(Addr_VCC_3194,0x3F,0x00);//Endless time

   I2C_WriteByte(Addr_VCC_3194,0x1C,0x01);//1 color enable
    I2C_WriteByte(Addr_VCC_3194,0x2C,0x01);//1 color enable
    I2C_WriteByte(Addr_VCC_3194,0x3C,0x01);//1 color enable
   I2C_WriteByte(Addr_VCC_3194,0x40,0xC5);//update color

   while(G_Demo_NO==1)
   {
      I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);//update p1
      while(G_Demo_NO==1)
      {
            if(I2C_ReadByte(Addr_VCC_3194,0x0D)==0x91)//P1 Running at T1
            {
            break;
          }
     }
      I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//update p2
      while(G_Demo_NO==1)
      {
            if(I2C_ReadByte(Addr_VCC_3194,0x0E)==0x91)//P2 Running at T1
            {
                  break;
            }
      }
      I2C_WriteByte(Addr_VCC_3194,0x43,0xC5);//update P3
      while(G_Demo_NO==1)
      {
            if(I2C_ReadByte(Addr_VCC_3194,0x0F)==0x91)//P3 Running at T1
            {
                  break;
            }
      }
      while(G_Demo_NO==1);
   }
```

MODE 2

T1 = T3 = 1.04S, T2 = T4 = 0.03S



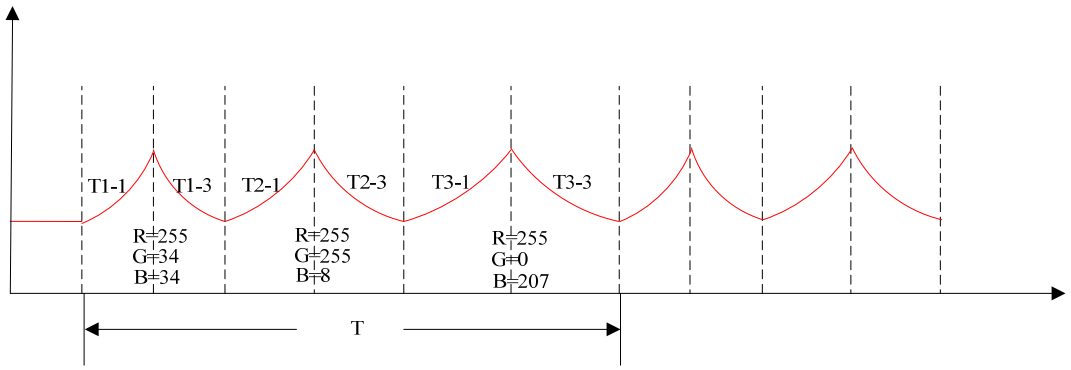

```
//Init
   I2C_WriteByte(Addr_VCC_3194,0x01,0x71);//current single mode, normal operation
   I2C_WriteByte(Addr_VCC_3194,0x02,0x07);//channel enable
   I2C_WriteByte(Addr_VCC_3194,0x03,0x3F);//chx max current 10mA 10mA 10mA
   I2C_WriteByte(Addr_VCC_3194,0x04,0x00);// Hold function disable
//pattern 1 color
   I2C_WriteByte(Addr_VCC_3194,0x10,0xff);// color 1
   I2C_WriteByte(Addr_VCC_3194,0x11,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x12,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x13,0xff);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x14,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x15,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x16,0xff);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x17,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x18,0xff);
//pattern 2 color
   I2C_WriteByte(Addr_VCC_3194,0x20,0xff);// color 1
   I2C_WriteByte(Addr_VCC_3194,0x21,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x22,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x23,0xff);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x24,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x25,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x26,0xff);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x27,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x28,0xff);
//pattern 3 color
   I2C_WriteByte(Addr_VCC_3194,0x30,0xff);// color 1
   I2C_WriteByte(Addr_VCC_3194,0x31,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x32,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x33,0xff);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x34,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x35,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x36,0xff);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x37,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x38,0xff);
//Pattern 1 timing
   I2C_WriteByte(Addr_VCC_3194,0x19,0x60);//T1 = 1.04, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1A,0x60);//T2 = 1.04s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1B,0x00);//T4&TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1D,0x15);//color cycle 1 time
   I2C_WriteByte(Addr_VCC_3194,0x1E,0x10);//Gamma=2.4, multy-pulse 1 time
   I2C_WriteByte(Addr_VCC_3194,0x1F,0x01);//pattern loop time
//Pattern 2 timing
   I2C_WriteByte(Addr_VCC_3194,0x29,0x60);//T1 = 1.04, Ts = 0.03s
```

```
   I2C_WriteByte(Addr_VCC_3194,0x2A,0x60);//T2 = 1.04s, T3 = 0.03s
    I2C_WriteByte(Addr_VCC_3194,0x2B,0x00);//T4&TP= 0.03s
    I2C_WriteByte(Addr_VCC_3194,0x2D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x2E,0x10);//Gamma=2.4, multy-pulse 1 time
    I2C_WriteByte(Addr_VCC_3194,0x2F,0x01);//pattern loop time
// Pattern 3 timing
   I2C_WriteByte(Addr_VCC_3194,0x39,0x60);//T1 = 1.04, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3A,0x60);//T2 = 1.04s, T3 = 0.03s
    I2C_WriteByte(Addr_VCC_3194,0x3B,0x00);//T4&TP= 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x3E,0x10);//Gamma=2.4, multy-pulse 1 time
    I2C_WriteByte(Addr_VCC_3194,0x3F,0x01);//pattern loop time

   I2C_WriteByte(Addr_VCC_3194,0x1C,0x01);//1 color enable
    I2C_WriteByte(Addr_VCC_3194,0x2C,0x01);//1 color enable
    I2C_WriteByte(Addr_VCC_3194,0x3C,0x01);//1 color enable

   I2C_WriteByte(Addr_VCC_3194,0x40,0xC5);//update color

   while(G_Demo_NO==2)
 {
      I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);// update p1
      while(I2C_ReadByte(Addr_VCC_3194,0x0D)!=0x00)// waiting p1 end
      {
            if(G_Demo_NO!=2)
            {
                  break;
            }
      }
      I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//update p2
      while(I2C_ReadByte(Addr_VCC_3194,0x0E)!=0x00)//waiting p2 end
      {
            if(G_Demo_NO!=2)
            {
                  break;
            }
      }
      I2C_WriteByte(Addr_VCC_3194,0x43,0xC5);// update p3
      while(I2C_ReadByte(Addr_VCC_3194,0x0F)!=0x00)//waiting p3 end
     {
            if(G_Demo_NO!=2)
            {
                  break;
            }
      }
      I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);// update p2
      while(I2C_ReadByte(Addr_VCC_3194,0x0E)!=0x00)//     waiting p2 end
      {
            if(G_Demo_NO!=2)
            {
                  break;
            }
      }
      I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);// update p1
      while(I2C_ReadByte(Addr_VCC_3194,0x0D)!=0x00)//waiting p1 end
      {
            if(G_Demo_NO!=2)
            {
                  break;
            }
```

```
    }
    I2C_WriteByte(Addr_VCC_3194,0x04,0x3F);// all hold on t2
    I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);// update p1
    while((I2C_ReadByte(Addr_VCC_3194,0x0D)&0x02)!=0x02)//wait hold on t2
    {
        if(G_Demo_NO!=2)
        {
            break;
        }
    }
    I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//    update p2
    while((I2C_ReadByte(Addr_VCC_3194,0x0E)&0x02)!=0x02)// wait hold on t2
    {
        if(G_Demo_NO!=2)
        {
            break;
        }
    }
    I2C_WriteByte(Addr_VCC_3194,0x43,0xC5);     //    update p3
    while((I2C_ReadByte(Addr_VCC_3194,0x0F)&0x02)!=0x02)// wait hold on t2
    {
        if(G_Demo_NO!=2)
        {
            break;
        }
    }
    I2C_WriteByte(Addr_VCC_3194,0x04,0x00);//clear hold on
    while(I2C_ReadByte(Addr_VCC_3194,0x0D)!=0x00)//wait all off
    {
        if(G_Demo_NO!=2)
        {
            break;
        }
    }
}
```

MODE 3

T1-1 = T1-3 = 0.26S, T2-1 = T2-3 = 0.38S, T3-1 = T3-3 = 0.51S

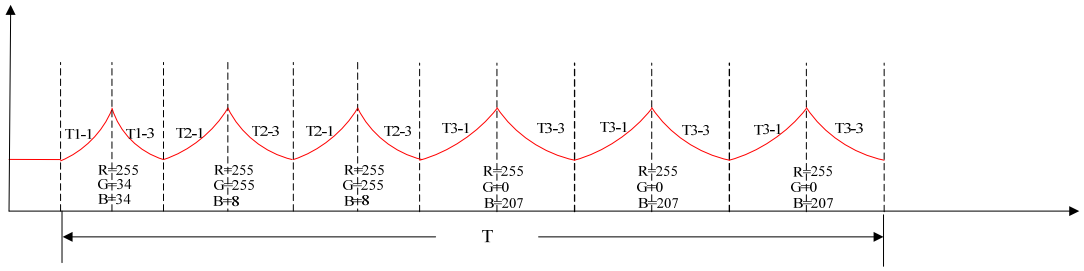



```
/Init:
   I2C_WriteByte(Addr_VCC_3194,0x01,0x75);//RGB mode, normal operation
   I2C_WriteByte(Addr_VCC_3194,0x02,0x07);//channel enable
   I2C_WriteByte(Addr_VCC_3194,0x03,0x3F);//chx max current 10mA 10mA 10mA
   I2C_WriteByte(Addr_VCC_3194,0x04,0x00);// Hold function disable
//pattern 1 color
   I2C_WriteByte(Addr_VCC_3194,0x10,34);//   color 1 Red
   I2C_WriteByte(Addr_VCC_3194,0x11,255);
   I2C_WriteByte(Addr_VCC_3194,0x12,34);
   I2C_WriteByte(Addr_VCC_3194,0x13,225);// color 2 no use
   I2C_WriteByte(Addr_VCC_3194,0x14,255);
   I2C_WriteByte(Addr_VCC_3194,0x15,8);
   I2C_WriteByte(Addr_VCC_3194,0x16,0);// color 3     no use
   I2C_WriteByte(Addr_VCC_3194,0x17,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x18,207);
//pattern 2 color
   I2C_WriteByte(Addr_VCC_3194,0x20,255);// color 1 Yellow
   I2C_WriteByte(Addr_VCC_3194,0x21,255);
   I2C_WriteByte(Addr_VCC_3194,0x22,8);
   I2C_WriteByte(Addr_VCC_3194,0x23,0xff);// color 2 no use
   I2C_WriteByte(Addr_VCC_3194,0x24,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x25,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x26,0xff);// color 3 no use
   I2C_WriteByte(Addr_VCC_3194,0x27,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x28,0xff);
//pattern 3 color
   I2C_WriteByte(Addr_VCC_3194,0x30,0);// color 1 purple
   I2C_WriteByte(Addr_VCC_3194,0x31,255);
   I2C_WriteByte(Addr_VCC_3194,0x32,207);
   I2C_WriteByte(Addr_VCC_3194,0x33,0xff);// color 2 no use
   I2C_WriteByte(Addr_VCC_3194,0x34,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x35,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x36,0xff);// color 3 no use
   I2C_WriteByte(Addr_VCC_3194,0x37,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x38,0xff);
//Pattern 1 timing
   I2C_WriteByte(Addr_VCC_3194,0x19,0x20);//T1 = 0.26s, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1A,0x20);//T2= 0.26s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1D,0x15);//color cycle 1 time
   I2C_WriteByte(Addr_VCC_3194,0x1E,0x11);//multy-pulse 1time ,next go to pattern 2
   I2C_WriteByte(Addr_VCC_3194,0x1F,0x01);//pattern 1 time
// Pattern 2 timing
   I2C_WriteByte(Addr_VCC_3194,0x29,0x30);//T1 = 0.38 , Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x2A,0x30);//T2= 0.38 , T4= 0.03s
```

```
   I2C_WriteByte(Addr_VCC_3194,0x2B,0x00);//T4 = TP = 0.03s
    I2C_WriteByte(Addr_VCC_3194,0x2D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x2E,0x12);//multy-pulse 1time ,next go to pattern 3
    I2C_WriteByte(Addr_VCC_3194,0x2F,0x01);//pattern 1 time
// Pattern 3 timing
   I2C_WriteByte(Addr_VCC_3194,0x39,0x40);//T1 = 0.51s, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3A,0x40);//T2 = 0.51s, T4= 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x3E,0x11);//multy-pulse 1time ,next go to pattern 1
    I2C_WriteByte(Addr_VCC_3194,0x3F,0x01);//pattern 1 time

   I2C_WriteByte(Addr_VCC_3194,0x1C,0x01);//1 color enable
    I2C_WriteByte(Addr_VCC_3194,0x2C,0x01);//1 color enable
    I2C_WriteByte(Addr_VCC_3194,0x3C,0x01);//1 color enable

    I2C_WriteByte(Addr_VCC_3194,0x40,0xC5);//update color
    I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);//update  P1
    I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//update  P2
    I2C_WriteByte(Addr_VCC_3194,0x43,0xC5);//update  P3
    while(G_Demo_NO==3);
```

MODE 4

T1-1 = T1-3 = 0.26S, T2-1 = T2-3 = 0.38S, T3-1 = T3-3 = 0.51S

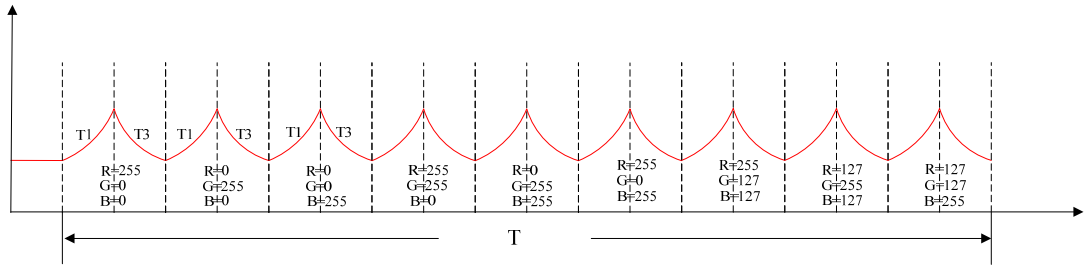



```
//Init
   I2C_WriteByte(Addr_VCC_3194,0x01,0x75);//RGB mode, normal operation
   I2C_WriteByte(Addr_VCC_3194,0x02,0x07);//channel enable
   I2C_WriteByte(Addr_VCC_3194,0x03,0x3F);//chx max current 10mA 10mA 10mA
   I2C_WriteByte(Addr_VCC_3194,0x04,0x00);// Hold function disable
//pattern 1 color
   I2C_WriteByte(Addr_VCC_3194,0x10,34);//   color 1 Red
   I2C_WriteByte(Addr_VCC_3194,0x11,255);
   I2C_WriteByte(Addr_VCC_3194,0x12,34);
   I2C_WriteByte(Addr_VCC_3194,0x13,225);// color 2 no use
   I2C_WriteByte(Addr_VCC_3194,0x14,255);
   I2C_WriteByte(Addr_VCC_3194,0x15,8);
   I2C_WriteByte(Addr_VCC_3194,0x16,0);// color 3     no use
   I2C_WriteByte(Addr_VCC_3194,0x17,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x18,207);
//pattern 2 color
   I2C_WriteByte(Addr_VCC_3194,0x20,255);// color 1 Yellow
   I2C_WriteByte(Addr_VCC_3194,0x21,255);
   I2C_WriteByte(Addr_VCC_3194,0x22,8);
   I2C_WriteByte(Addr_VCC_3194,0x23,0xff);// color 2 no use
   I2C_WriteByte(Addr_VCC_3194,0x24,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x25,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x26,0xff);// color 3 no use
   I2C_WriteByte(Addr_VCC_3194,0x27,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x28,0xff);
//pattern 3 color
   I2C_WriteByte(Addr_VCC_3194,0x30,0);// color 1 purple
   I2C_WriteByte(Addr_VCC_3194,0x31,255);
   I2C_WriteByte(Addr_VCC_3194,0x32,207);
   I2C_WriteByte(Addr_VCC_3194,0x33,0xff);// color 2 no use
   I2C_WriteByte(Addr_VCC_3194,0x34,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x35,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x36,0xff);// color 3 no use
   I2C_WriteByte(Addr_VCC_3194,0x37,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x38,0xff);
//Pattern 1 timing
   I2C_WriteByte(Addr_VCC_3194,0x19,0x30);//T1 = 0.26s, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1A,0x30);//T2 = 0.26s, T3 = 0.03s
    I2C_WriteByte(Addr_VCC_3194,0x1B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1D,0x15);//color cycle 1 time
    I2C_WriteByte(Addr_VCC_3194,0x1E,0x11);//multy-pulse 1time ,next go to pattern 2
    I2C_WriteByte(Addr_VCC_3194,0x1F,0x01);//pattern 1 time
//Pattern 2 timing
   I2C_WriteByte(Addr_VCC_3194,0x29,0x50);//T1 = 0.38s, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x2A,0x50);//T2 = 0.38s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x2B,0x00);//T4 = TP = 0.03s
    I2C_WriteByte(Addr_VCC_3194,0x2D,0x16);//color cycle 2 time
   I2C_WriteByte(Addr_VCC_3194,0x2E,0x12);//multy-pulse 1time ,next go to pattern 3
```

```
   I2C_WriteByte(Addr_VCC_3194,0x2F,0x01);//pattern 1 time
//Pattern 3 timing
   I2C_WriteByte(Addr_VCC_3194,0x39,0x60);//T1 = 0.51s, Ts = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3A,0x60);//T2 = 0.51s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3D,0x17);//color cycle 3 time
   I2C_WriteByte(Addr_VCC_3194,0x3E,0x11);//multy-pulse 1time ,next go to pattern 1
   I2C_WriteByte(Addr_VCC_3194,0x3F,0x01);//pattern 1 time

   I2C_WriteByte(Addr_VCC_3194,0x1C,0x01);//1 color enable
   I2C_WriteByte(Addr_VCC_3194,0x2C,0x01);//1 color enable
   I2C_WriteByte(Addr_VCC_3194,0x3C,0x01);//1 color enable

   I2C_WriteByte(Addr_VCC_3194,0x40,0xC5);//update color
   I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);//update  P1
   I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//update  P2
   I2C_WriteByte(Addr_VCC_3194,0x43,0xC5);//update  P3
   while(G_Demo_NO==4);
```

MODE 5

T1 = T3 = 0.77S




```
//Init
   I2C_WriteByte(Addr_VCC_3194,0x01,0x75);//RGB mode, normal operation
   I2C_WriteByte(Addr_VCC_3194,0x02,0x07);//channel enable
   I2C_WriteByte(Addr_VCC_3194,0x03,0x3F);//chx max current 10mA 10mA 10mA
   I2C_WriteByte(Addr_VCC_3194,0x04,0x00);//Hold function disable
//pattern 1 color
   I2C_WriteByte(Addr_VCC_3194,0x10,00);//   color 1 Red
   I2C_WriteByte(Addr_VCC_3194,0x11,255);
   I2C_WriteByte(Addr_VCC_3194,0x12,00);
   I2C_WriteByte(Addr_VCC_3194,0x13,225);// color 2 green
   I2C_WriteByte(Addr_VCC_3194,0x14,0);
   I2C_WriteByte(Addr_VCC_3194,0x15,0);
   I2C_WriteByte(Addr_VCC_3194,0x16,0);// color 3     blue
   I2C_WriteByte(Addr_VCC_3194,0x17,0);
   I2C_WriteByte(Addr_VCC_3194,0x18,255);
//pattern 2 color
   I2C_WriteByte(Addr_VCC_3194,0x20,255);// color 1 yellow
   I2C_WriteByte(Addr_VCC_3194,0x21,255);
   I2C_WriteByte(Addr_VCC_3194,0x22,0);
   I2C_WriteByte(Addr_VCC_3194,0x23,0xff);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x24,0);
   I2C_WriteByte(Addr_VCC_3194,0x25,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x26,0);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x27,0xff);
   I2C_WriteByte(Addr_VCC_3194,0x28,0xff);
//pattern 3 color
   I2C_WriteByte(Addr_VCC_3194,0x30,127);// color 1
   I2C_WriteByte(Addr_VCC_3194,0x31,255);
   I2C_WriteByte(Addr_VCC_3194,0x32,127);
   I2C_WriteByte(Addr_VCC_3194,0x33,0xff);// color 2
   I2C_WriteByte(Addr_VCC_3194,0x34,127);
   I2C_WriteByte(Addr_VCC_3194,0x35,127);
   I2C_WriteByte(Addr_VCC_3194,0x36,127);// color 3
   I2C_WriteByte(Addr_VCC_3194,0x37,127);
   I2C_WriteByte(Addr_VCC_3194,0x38,0xff);
//Pattern 1 timing
   I2C_WriteByte(Addr_VCC_3194,0x19,0x50);//T1 = 0.77s, TS = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1A,0x50);//T2= 0.77s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x1D,0x15);//color cycle 1 time
   I2C_WriteByte(Addr_VCC_3194,0x1E,0x11);//multy-pulse 1time ,next go to pattern 2
   I2C_WriteByte(Addr_VCC_3194,0x1F,0x01);//pattern 1 time
//Pattern 2 timing
   I2C_WriteByte(Addr_VCC_3194,0x29,0x50);//T1 = 0.77s, TS = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x2A,0x50);//T2= 0.77s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x2B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x2D,0x15);//color cycle 1 time
   I2C_WriteByte(Addr_VCC_3194,0x2E,0x12);//multy-pulse 1time ,next go to pattern 3
```

```
   I2C_WriteByte(Addr_VCC_3194,0x2F,0x01);//pattern 1 time
//Pattern 3 timing
   I2C_WriteByte(Addr_VCC_3194,0x39,0x50);//T1 = 0.77s, TS = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3A,0x50);//T2= 0.77s, T3 = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3B,0x00);//T4 = TP = 0.03s
   I2C_WriteByte(Addr_VCC_3194,0x3D,0x15);//color cycle 1 time
   I2C_WriteByte(Addr_VCC_3194,0x3E,0x11);//multy-pulse 1time ,next go to pattern 1
   I2C_WriteByte(Addr_VCC_3194,0x3F,0x01);//pattern 1 time

   I2C_WriteByte(Addr_VCC_3194,0x1C,0x07);//3 color enable
   I2C_WriteByte(Addr_VCC_3194,0x2C,0x07);//3 color enable
   I2C_WriteByte(Addr_VCC_3194,0x3C,0x07);//3 color enable

   I2C_WriteByte(Addr_VCC_3194,0x40,0xC5);//update color
   I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);//update  P1
   I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//update  P2
   I2C_WriteByte(Addr_VCC_3194,0x43,0xC5);//update  P3
   while(G_Demo_NO==5);
```

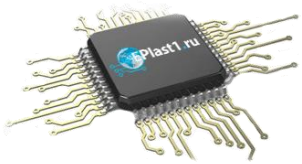
Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный)
Факс: 8 (812) 320-02-42
Электронная почта: org@eplast1.ru
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.