

3-CHANNEL LED DRIVER

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

- $\leq 5.5V$ 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.

3-CHANNEL LED DRIVER

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 board's MCU (3.0V) and maybe damaged.

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

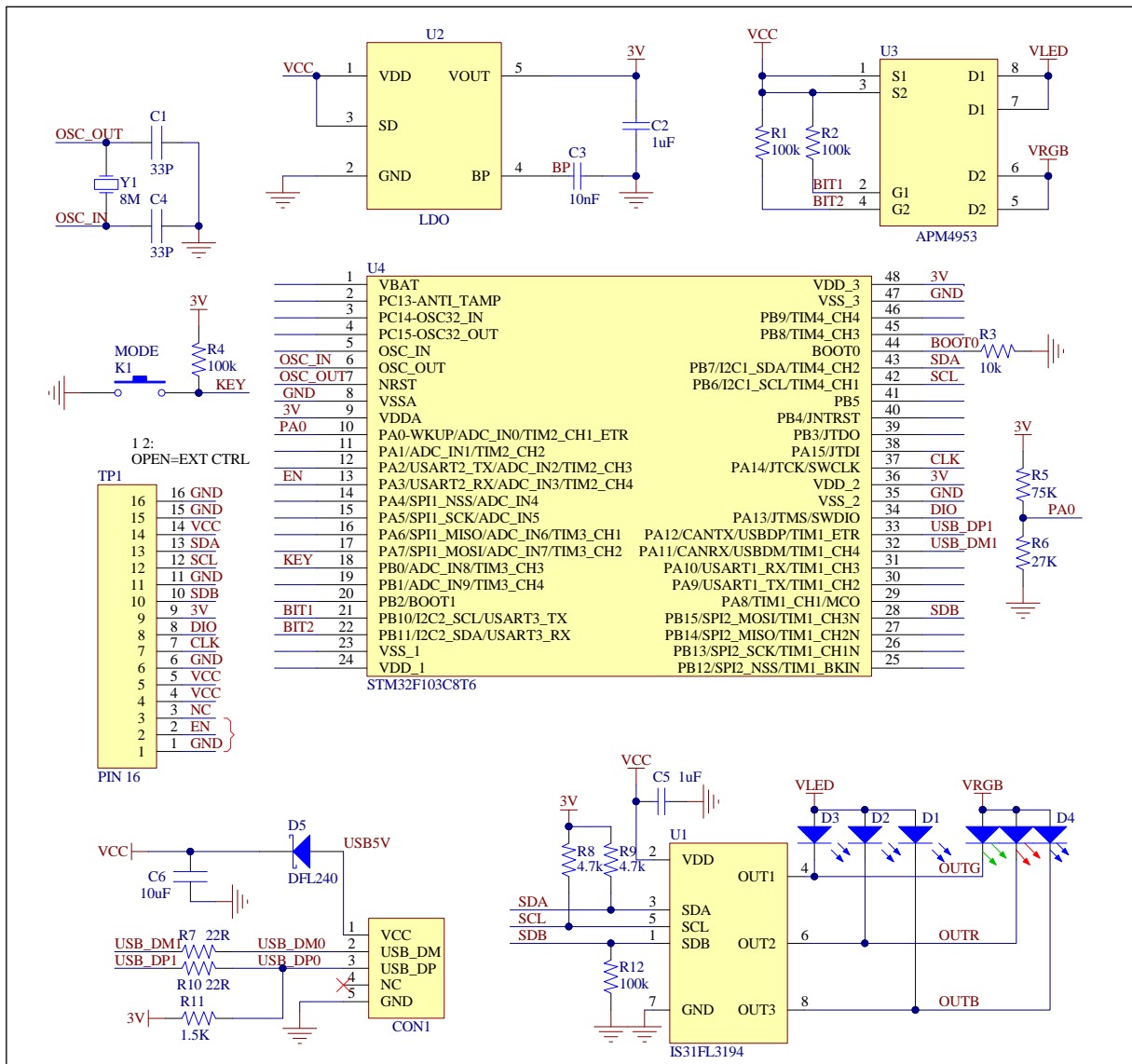


Figure 2: IS32FL3194 Application Schematic

3-CHANNEL LED DRIVER

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.

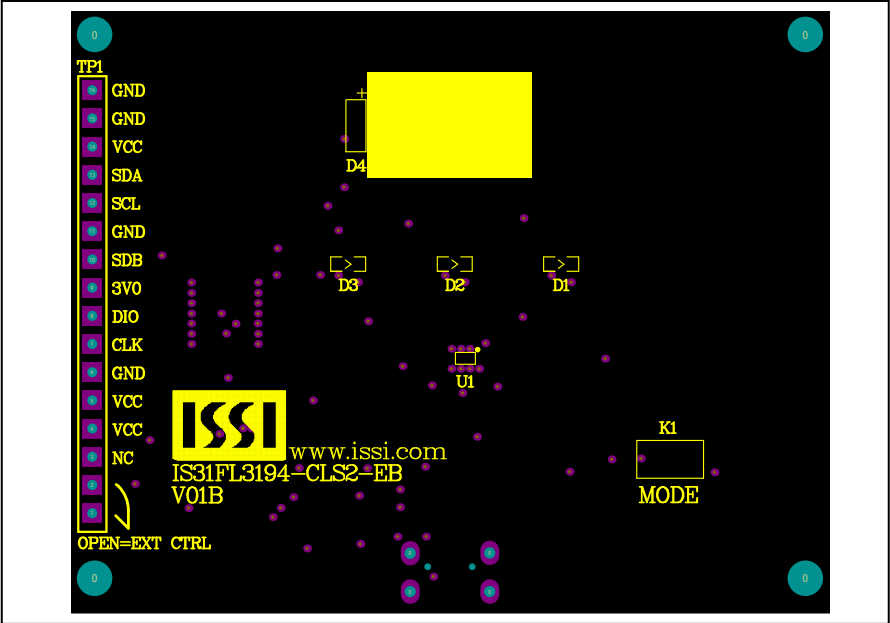


Figure 3: Board Component Placement Guide - Top Layer

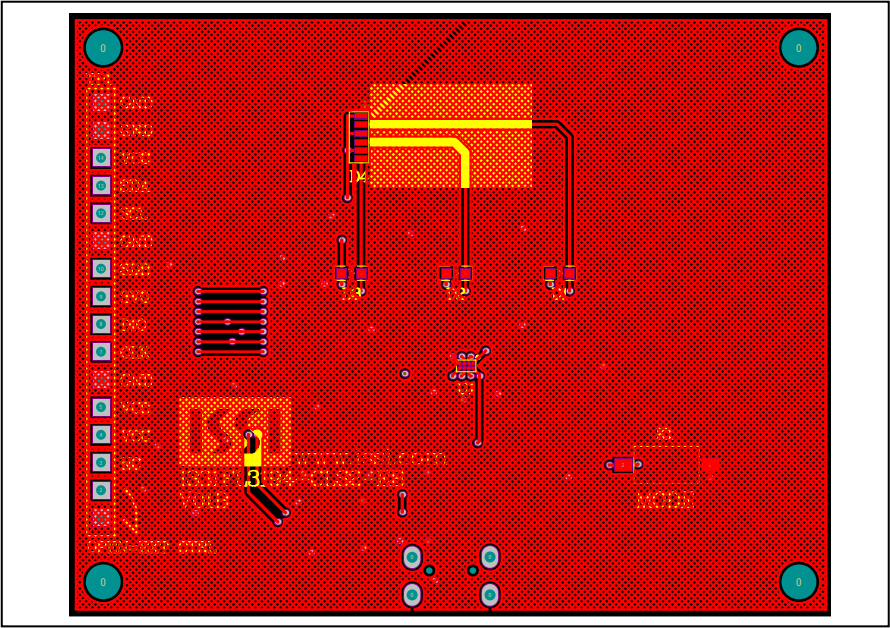


Figure 4: Board PCB Layout - Top Layer

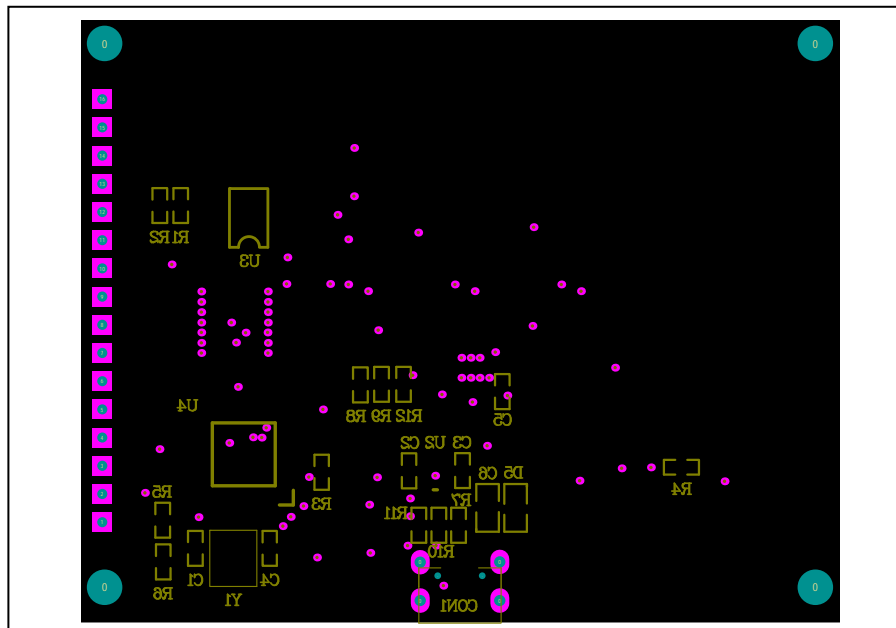


Figure 5: Board Component Placement Guide - Bottom Layer

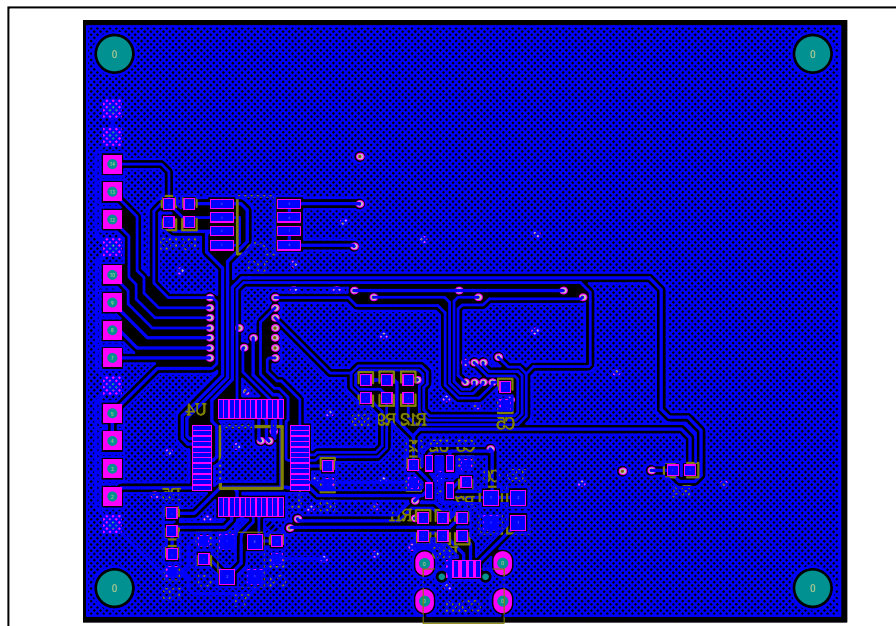


Figure 6: Board PCB Layout - Bottom Layer

Copyright © 2017 Integrated Silicon Solution, Inc. All rights reserved. ISSI reserves the right to make changes to this specification and its products at any time without notice. ISSI assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products. Integrated Silicon Solution, Inc. does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless Integrated Silicon Solution, Inc. receives written assurance to its satisfaction, that:

- a.) the risk of injury or damage has been minimized;
- b.) the user assume all such risks; and
- c.) potential liability of Integrated Silicon Solution, Inc is adequately protected under the circumstances



3-CHANNEL LED DRIVER

REVISION HISTORY

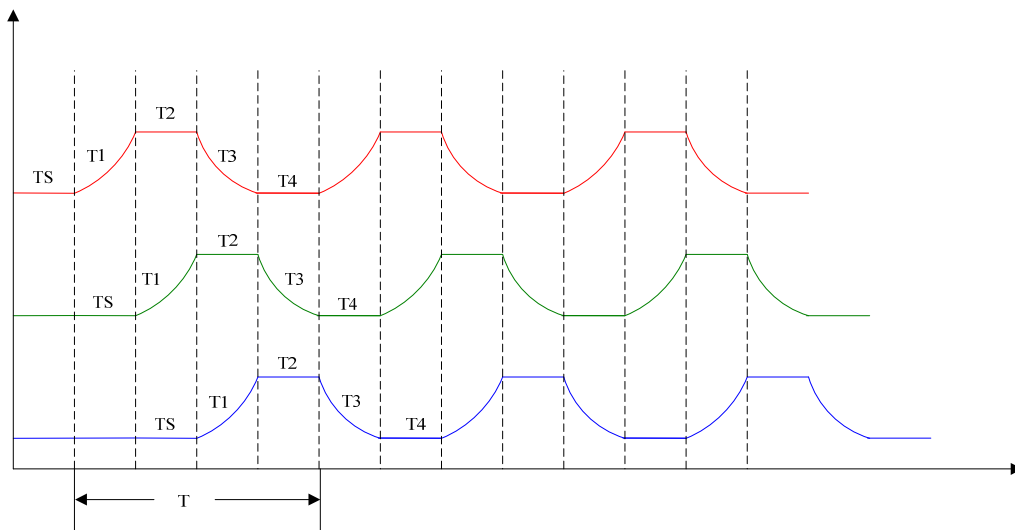
Revision	Detail Information	Date
A	Initial release	2017.03.23
B	Add appendix	2017.08.09

3-CHANNEL LED DRIVER

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
```

3-CHANNEL LED DRIVER

```

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

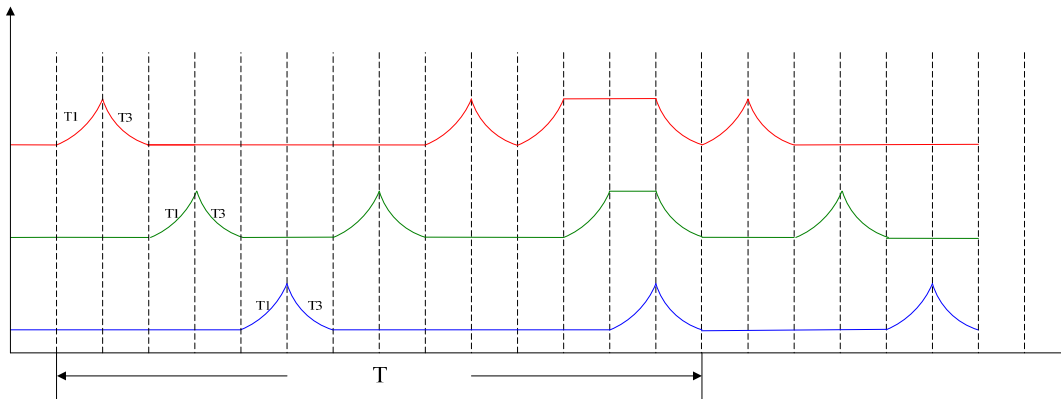
while(G_Demo_NO==1)
{
    I2C_WriteByte(Addr_VCC_3194,0x41,0xC5);//update p1
    while(G_Demo_NO==1)
    {
        if(I2C_ReadByte(Addr_VCC_3194,0x0D)==0x91)//P1 Running at T1
        {
            break;
        }
    }
    I2C_WriteByte(Addr_VCC_3194,0x42,0xC5);//update p2
    while(G_Demo_NO==1)
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

```


3-CHANNEL LED DRIVER

MODE 2

$T_1 = T_3 = 1.04S$, $T_2 = T_4 = 0.03S$



```
//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
```

3-CHANNEL LED DRIVER

```

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;
        }
    }
}

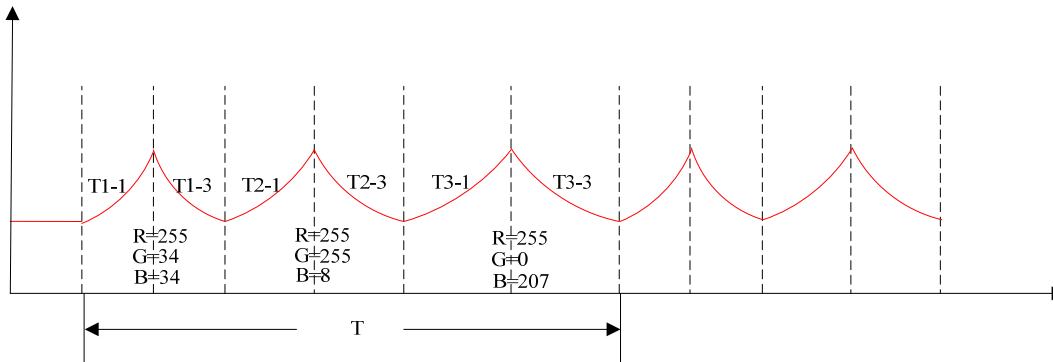
```

```
}
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;
    }
}
}
```

3-CHANNEL LED DRIVER

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
```

3-CHANNEL LED DRIVER

```
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

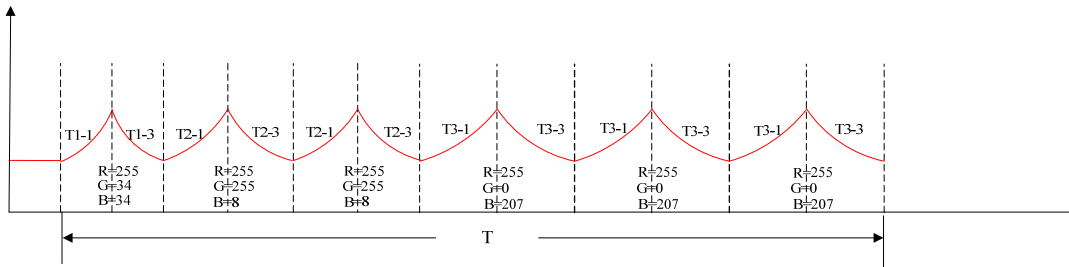
I2C_WriteByte(Addr_VCC_3194,0x1C,0x01);//1 color enable
I2C_WriteByte(Addr_VCC_3194,0x2C,0x01);//1 color enable
I2C_WriteByte(Addr_VCC_3194,0x3C,0x01);//1 color enable

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);
```

3-CHANNEL LED DRIVER

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
```

3-CHANNEL LED DRIVER

```
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

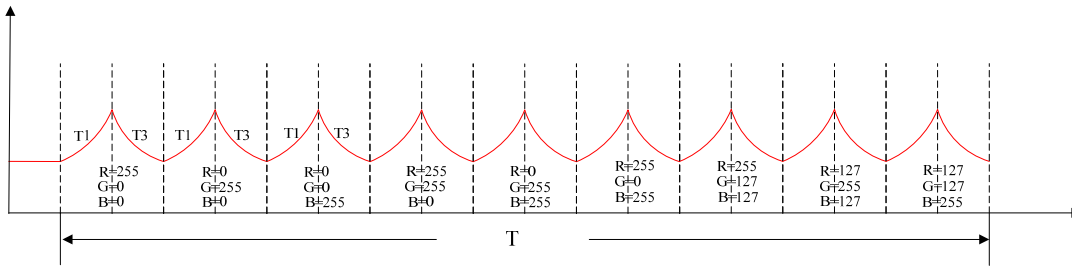
I2C_WriteByte(Addr_VCC_3194,0x1C,0x01);//1 color enable
I2C_WriteByte(Addr_VCC_3194,0x2C,0x01);//1 color enable
I2C_WriteByte(Addr_VCC_3194,0x3C,0x01);//1 color enable

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);
```

3-CHANNEL LED DRIVER

MODE 5

$T_1 = T_3 = 0.77S$



```
//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
```


3-CHANNEL LED DRIVER

```
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);
```

Данный компонент на территории Российской Федерации

Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

<http://moschip.ru/get-element>

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:

moschip.ru

moschip.ru_4

moschip.ru_6

moschip.ru_9