CHARGE PUMP RGBW LED DRIVER WITH PROGRAMMABLE SELF-RUNNING PATTERNS EVALUATION BOARD GUIDE

DESCRIPTION

The IS31FL3195 is a compact and efficient 4-channel charge pump LED driver with programmable sequence operation for automated RGBW lighting effects. It is capable of driving 1 to 4 LEDs with a low drop-out and current matching so all 4 LEDs maintain consistent brightness. Each channel can support up to 20mA of current.

The built−in charge pump (CP) structure will automatically toggle between 1x, 1.5x operation depending on the battery's state of charge. This DC/DC converter operates at a high switching frequency which enables the use of small external capacitors and achieves 92% peak total efficiency. The IS31FL3195 is optimized for battery applications. To conserve battery life, the charge pump goes into high impedance mode whenever the IS31FL3195 is shutdown consuming less than 1μA.

The IS31FL3195 can operate in either "Current Level" or "Programmable Sequence" mode. In Current Level mode, the average output current of each output is independently programmed and controlled in 256 PWM steps to simplify color mixing. In programmable sequence mode, the timing characteristics for each output can be individually adjusted to maintain a pre-established pattern sequence without requiring any additional MCU interaction, thus saving valuable system resources.

QUICK START

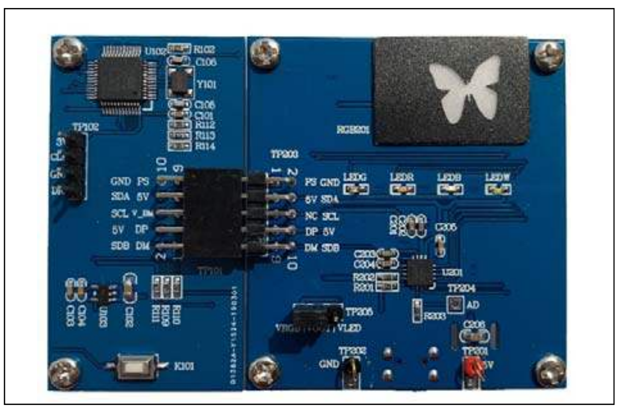

Figure 1: Photo of IS31FL3195-QFLS2 Evaluation Board

FEATURES

- 2.7V to 5.5V supply voltage
- Charge pump
  - 1x, 1.5x operating modes
  - Highly efficient across battery state of charge
  - 1MHz constant frequency
  - Enter 1.5x mode if any of the four OUTx pins < 150mV (50mV~300mV selectable)
- Power saving operating
  - 1μA shutdown current
  - 640μA quiescent operating current (CP1x mode)- Two selectable sleep modes
    - Sleep 1 Mode: 38μA
    - Sleep 2 Mode: 1μA
  - Auto sleep mode if all OUTx outputs are off for >30s.
- Support four LEDs - RGBW
  - Resistor sets LED current up to 20mA
  - 8-bit dot correction
- LEDs can operate with pre-established lighting patterns
  - Run without a micro
  - Fixed number of iterations or non-stop operation
  - Each channel has its own fade registers (TS-T4) with independent start/stop
- 1MHz I2C bus interface
  - Automatic address increment function
  - 4 selectable I2C address locations
- Over-temperature protection
- QFN-16 (4mm × 4mm) packages
- Operating temperature T_A= -40°C ~ +85°C

RECOMMENDED EQUIPMENT

- 5.0V, 2A power supply

ABSOLUTE MAXIMUM RATINGS

- ≤ 5.5V power supply

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

ORDERING INFORMATION

Part No.	Temperature Range	Package
IS31FL3195-QFLS2-EB	-40°C to +85°C, Industrial	QFN-16, Lead-free

Table 1: Ordering Information

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

PROCEDURE

The IS31FL3195 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) Connect the MCU board's TP101 to the IS31FL3195 board's TP203 to enable the control of MCU (default status).
2) Connect the Variable power supply to VIN/GND, or plug in the USB power input to micro-USB.
3) Short VOUT and VLED in TP205 to enable the single color LEDs, or short VOUT and VRGB in TP205 to enable the RGB LED.
4) 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 IS31FL3195 evaluation board has 4 display modes. Press MODE (K101) button to switch configurations.

1) (Default mode) Single color LED mode.
2) RGB pattern+ White LED mode.
3) RGBY pattern mode.
4) Single mode + OUT4 DC mode.
5) Disconnect TP101 from TP203 (remove the MCU control board), the IS31FL3195 board will continue running the K101 selected pattern.
6) If a variable power supply is connect to TP201 and TP202, lower the voltage to 2.7V to verify continued operation under low voltage condition.

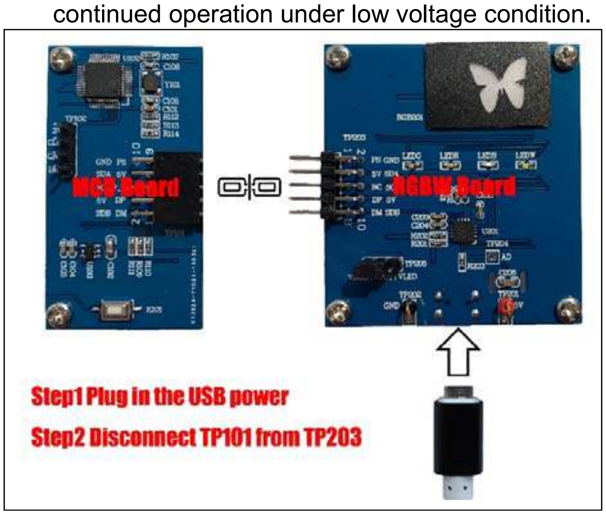

Figure 2: Photo of RGBW Board disconnected from MCU board

Note: IS31FL3195 solely controls the FxLED function on the evaluation board.

SOFTWARE SUPPORT

The MCU board's TP101 interconnects with the IS31FL3195 board's TP203. If TP101 is disconnected from TP203, the MCU to IS31FL3195 communication will be removed and the IS31FL3195 board will continue running the K101 selected pattern. The I2C pins are floated and SDB pin is pulled high by R201. An external I2C and SDB signal can be connected to TP203 to externally control the IS31FL3195 LED driver. Short TP205's VOUT to VLED to enable the individual LEDs (LEDG/LEDR/LEDB), or short VOUT to VRGB to enable the RGB Butterfly. LEDW is always ON and not controlled by TP205 jumper position.

The steps listed below are an example for using the Arduino as external control.

The Arduino hardware consists of an Atmel microcontroller with a bootloader allowing quick firmware updates. First download the latest Arduino Integrated Development Environment IDE (1.6.12 or greater) from www.arduino.cc/en/Main/Software. Also download the Wire.h library from www.arduino.cc/en/reference/wire and verify that pgmspace.h is in the directory …program Files(x86)/Arduino/hardware/tools/avr/avr/include/avr /. Then download the latest IS31FL3195 test firmware (sketch) from the Lumissil website http://www.lumissil.com/products/led-driver/fxled.

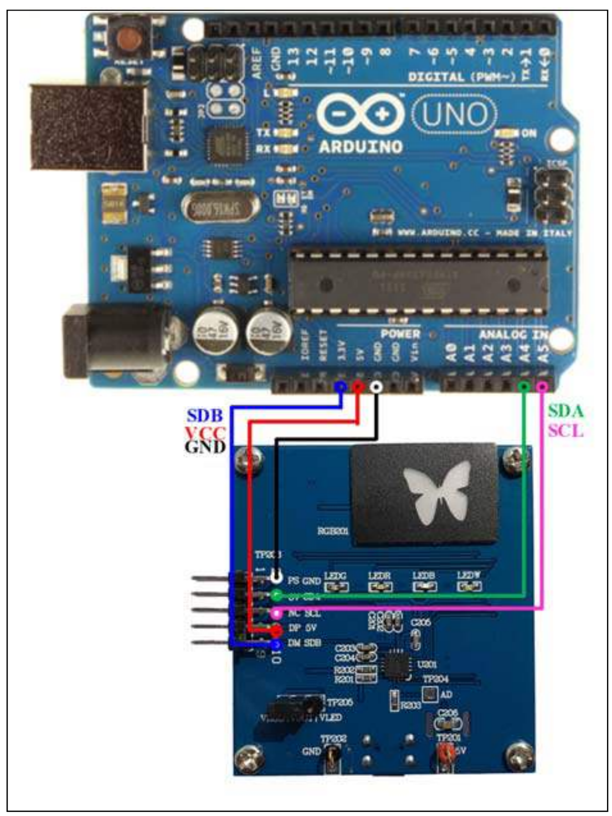

Figure 3: Photo of Arduino UNO connected to RGBW Board

1) TP101 disconnect from TP203 (Remove MCU

control board)

2) Connect the 5 pins from Arduino board to the RGBW Board:
   a) Arduino 5V pin to RGBW board VCC.
   b) Arduino GND to RGBW board GND.
   c) Arduino SDA (A4) to RGBW board SDA.
   d) Arduino SCL (A5) to RGBW board SCL.
   e) (Optional, SDB has a weak pull-up to 5V by R201) If Arduino uses a 3.3V MCU VCC, connect 3.3V to RGBW board SDB (TP203), if Arduino uses a 5.0V MCU VCC, connect 5.0V or 3.3V to RGBW board SDB (TP203). (Arduino UNO MCU VCC is 5V, so SDB can be 5V or 3.3V). I2C address can be changed by connecting TP204 to VCC, SCL or SDA.
3) Use the test code in appendix I or download the test firmware (sketch) from the Lumissil website, and copy the code to Arduino IDE, compile and upload to Arduino.
4) Run the Arduino code which will program the IS31FL3195 to cycle through color patterns, there is no button control.
5) Short VOUT and VLED in TP205 to enable the single color LEDs (LEDG/LEDR/LEDB), or short VOUT and VRGB in TP205 to enable the RGB LED. LEDW is always light up.

Please refer to the datasheet to get more information about IS31FL3195.

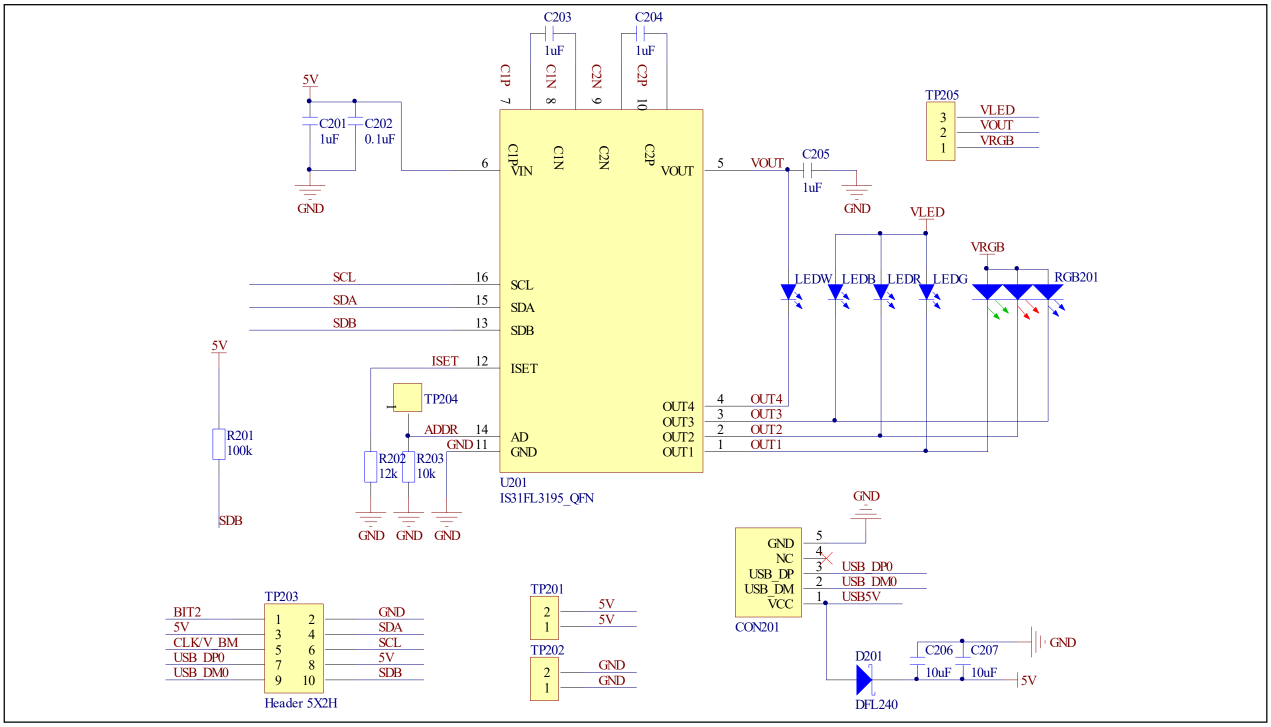

Figure 4: IS31FL3195 Schematic

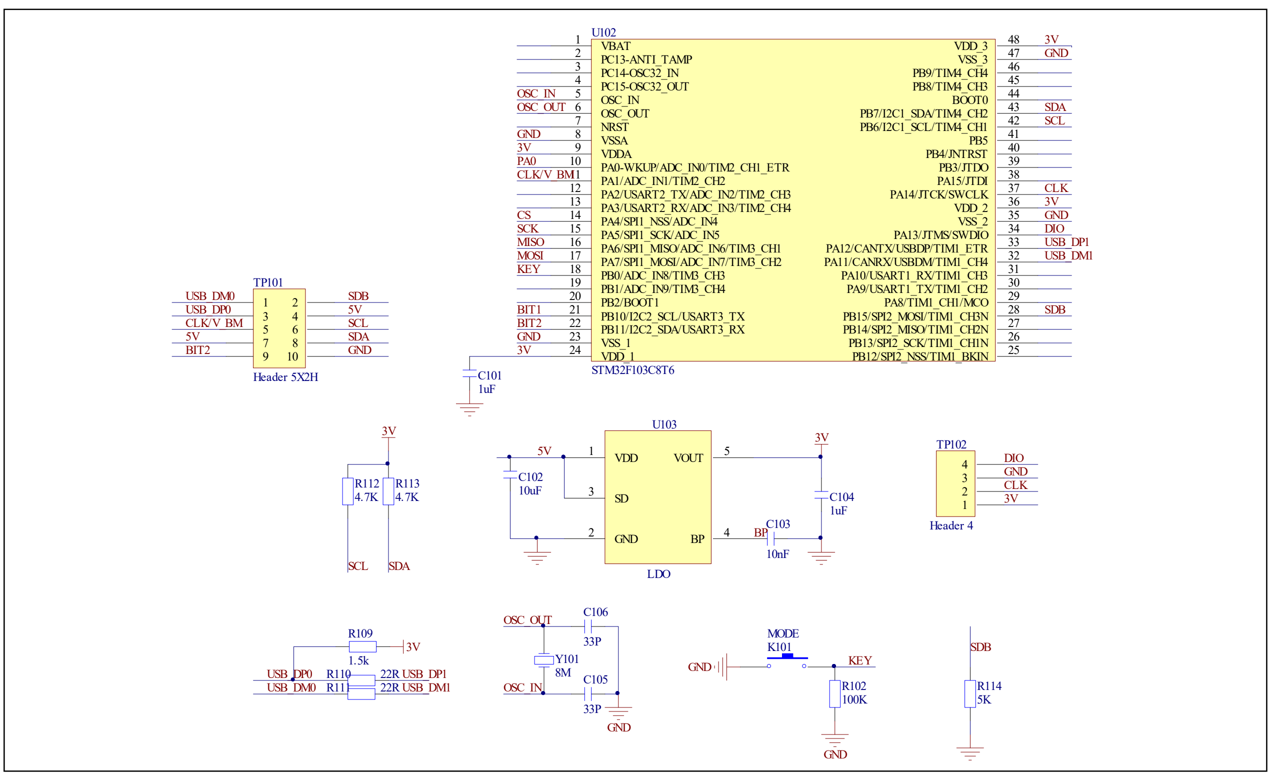

Figure 5: MCU Schematic

BILL OF MATERIALS

IS31FL3195

Name	Symbol	Description	Qty	Supplier	Part No.
LED Driver	U201	4CH FxLED Driver	1	Lumissil	IS31FL3195
Diode	D201	Diode, SMD	1	DIODES	DFLS240
LED	LEDB	Diode, LED Blue, SMD	1	Everlight	19-217/BHC-AN1P2/3T
LED	LEDG	Diode, LED Green, SMD	1	Everlight	19-217/GHC-YR1S2/3T
LED	LEDR	Diode, LED Red, SMD	1	Everlight	19-217/R6C-P1Q2/3T
LED	LEDW	Diode, LED White, SMD	1	Everlight	GT-197
LED	RGB201	Diode, LED RGB, SMD	1	Everlight	99-235/RSGBB7C-A22/2D or 99-235/RGBC/TR8
Resistor	R201	RES,100k,1/10W,±5%,SMD	3	Yageo	RC0603JR-07100KL
Resistor	R202	RES,12k,1/10W,±5%,SMD	1	Yageo	RC0603JR-0712KL
Resistor	R203	RES,10k,1/10W,±5%,SMD	2	Yageo	RC0603JR-0710KL
Capacitor	C201,C203, C204,C205	CAP, 1μF,16V,±10%,SMD	4	Yageo	CC0603KRX7R7BB105
Capacitor	C202	CAP,100nF,16V,±20%,SMD	1	Yageo	CC0603MRX7R7BB104
Capacitor	C206,C207	CAP,10μF,16V,±20%,SMD	1	Yageo	CC0805MRX5R7BB106
Header	TP203	5×2 Connection Jumper	1		
Header	TP204	ADDR Pin	1		
Header	TP205	RGB/LEDs select Jumper	1		
Diffuser		RGB Butterfly	1		Proprietary

Bill of Materials, refer to Figure 4 above.

STM32F103C8T6

Name	Symbol	Description	Qty	Supplier	Part No.
MCU	U102	Microcontroller	1	STM	STM32F103C8T6
LDO	U103	3.0V LDO	1	SGMICRO	SGM2019-3.0YN5G
Crystal	Y101	Crystal, 8MHz	1	HLX	HC-49S
Resistor	R102	RES,100k,1/10W,±5%,SMD	1	Yageo	RC0603JR-07100KL
Resistor	R109	RES,1.5k,1/10W,±5%,SMD	1	Yageo	RC0603JR-071K5L
Resistor	R110,R111	RES,22R,1/10W,±5%,SMD	2	Yageo	RC0603JR-0722RL
Resistor	R112,R113	RES,4.7k,1/10W,±5%,SMD	2	Yageo	RC0603JR-074K7L
Resistor	R114	RES,5k,1/10W,±5%,SMD	1	Yageo	RC0603JR-0705KL
Capacitor	C101,C104	CAP, 1μF,16V,±10%,SMD	2	Yageo	CC0603KRX7R7BB105
Capacitor	C102	CAP,10μF,16V,±20%,SMD	1	Yageo	CC0603MRX5R7BB106
Capacitor	C103	CAP,10nF,16V,±10%,SMD	1	Yageo	CC0603KPX7R7BB103
Capacitor	C105,C106	CAP,33pF,50V,±5%,SMD	2	Yageo	CQ0603JRNPO9BN360
Header	TP101	5×2 Connection Jumper	1		
Button	K101	Button SMD	1		

Bill of Materials, refer to Figure 5 above.

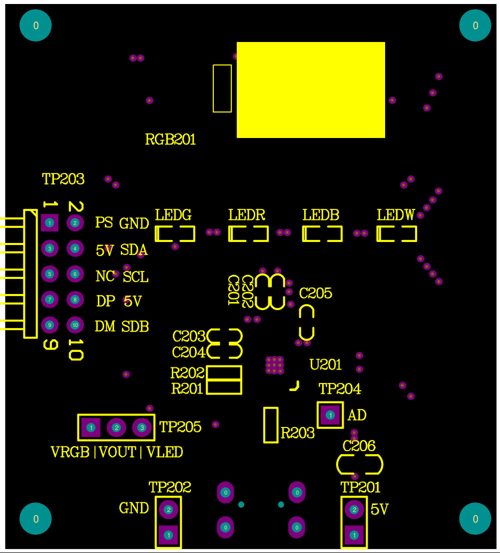

Figure 6: Board Component Placement Guide - Top Layer

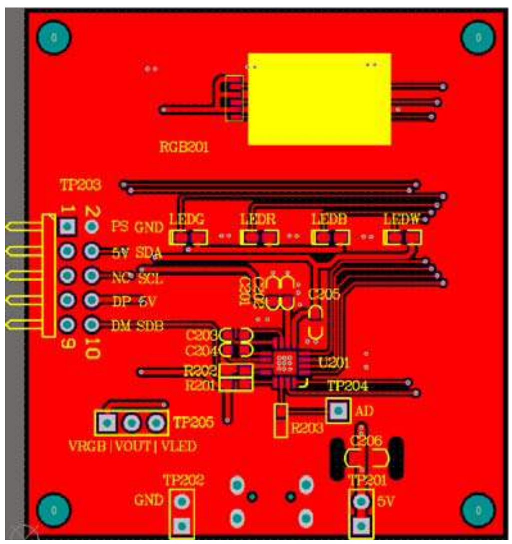

Figure 7: Board PCB Layout - Top Layer

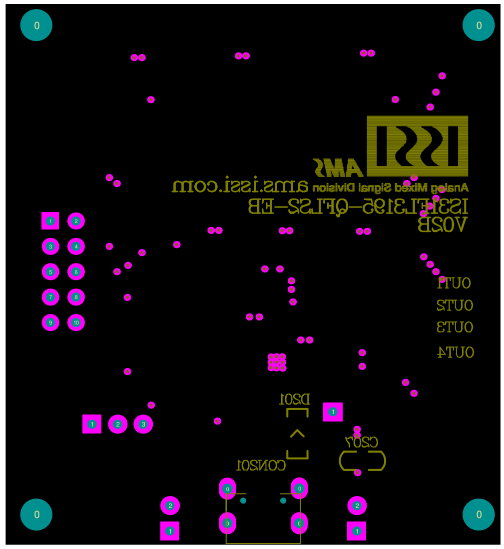

Figure 8: Board Component Placement Guide - Bottom Layer

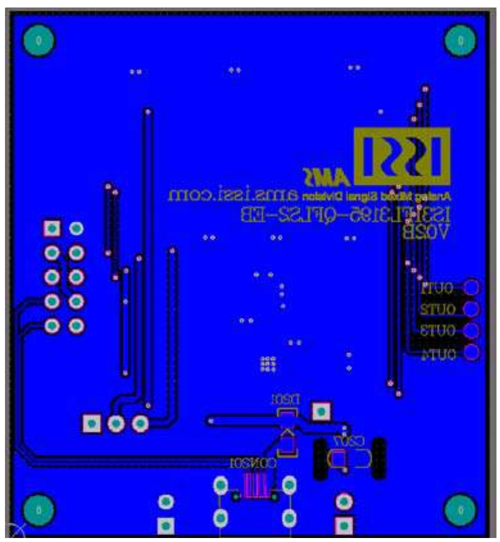

Figure 9: Board PCB Layout - Bottom Layer

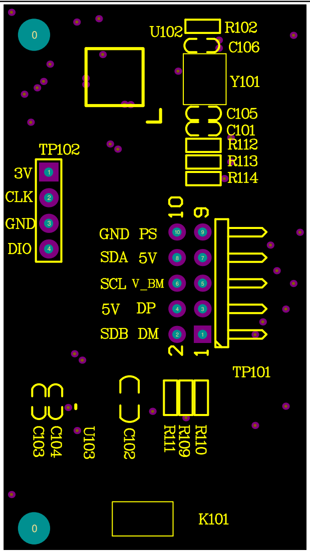

Figure 10: Board Component Placement Guide - Top Layer

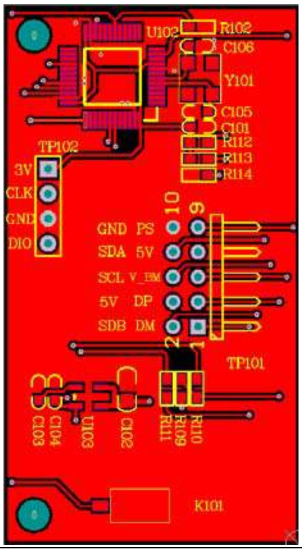

Figure 11: Board PCB Layout - Top Layer

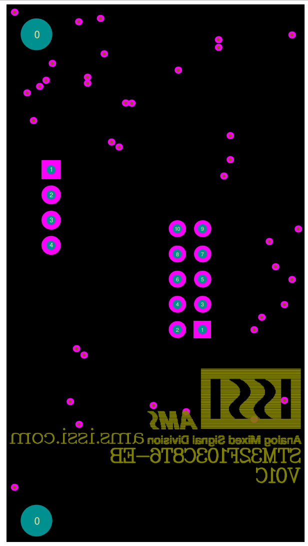

Figure 12: Board Component Placement Guide - Bottom Layer

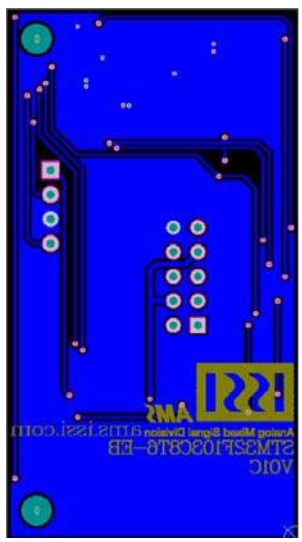

Figure 13: Board PCB Layout - Bottom Layer

REVISION HISTORY

Revision	Detail Information	Data
A	Initial release	2019.03.01
B	Update the BOM	2021.05.08

APPENDIX Ⅰ: IS31FL3195 Arduino Test Code V01A

```
#include<Wire.h>
#include<avr/pgmspace.h>
#define Addr_GND 0xA8 //Default RGBW board connection, Connect TP204 to SCL, SDA or VCC for different address

void setup() {

   // put your setup code here, to run once:
   Wire.begin();
   Wire.setClock(400000);//I2C 400kHz
//    pinMode(4,OUTPUT);//SDB
  // digitalWrite(4,HIGH);//SDB_HIGH
   //delay(100); //keep 0.5s
   Init_FL3195();
   IS31FL3195_mode1();//breath mode
}

void loop() {
   // put your main code here, to run repeatedly:
   // Init_FL3195();
}

void IS_IIC_WriteByte(uint8_t Dev_Add,uint8_t Reg_Add,uint8_t Reg_Dat)
{
   Wire.beginTransmission(Dev_Add/2);
   Wire.write(Reg_Add); // sends regaddress
   Wire.write(Reg_Dat); // sends regaddress
   Wire.endTransmission(); // stop transmitting
}

void Init_FL3195(void)
{
   IS_IIC_WriteByte(Addr_GND,0x01,0xF1);//normal operation
   IS_IIC_WriteByte(Addr_GND,0x02,0xF2);//LED mode choose
   IS_IIC_WriteByte(Addr_GND,0x03,0x00);//Charge Pump Setting-1
   IS_IIC_WriteByte(Addr_GND,0x05,0xFF);//Current Band Setting
}

void IS31FL3195_mode1(void)//white LED
{
   //P1 time and cycle
   IS_IIC_WriteByte(Addr_GND,0x1D,0x15);//Color Cycle Times
   IS_IIC_WriteByte(Addr_GND,0x1C,0x07);//P4 Color Enable
   IS_IIC_WriteByte(Addr_GND,0x19,0x44);//TS &T1 Setting
   IS_IIC_WriteByte(Addr_GND,0x1A,0x44);//T2 &T3 Setting
   IS_IIC_WriteByte(Addr_GND,0x1B,0x44);//TP &T4 Setting
   IS_IIC_WriteByte(Addr_GND,0x1E,0x11);//P1 NXT
   IS_IIC_WriteByte(Addr_GND,0x1F,0x01);//Loop Times

   //P2 time and cycle
   IS_IIC_WriteByte(Addr_GND,0x2D,0x15);//Color Cycle Times
   IS_IIC_WriteByte(Addr_GND,0x2C,0x07);//P4 Color Enable
   IS_IIC_WriteByte(Addr_GND,0x29,0x44);//TS &T1 Setting
   IS_IIC_WriteByte(Addr_GND,0x2A,0x44);//T2 &T3 Setting
   IS_IIC_WriteByte(Addr_GND,0x2B,0x44);//TP &T4 Setting
   IS_IIC_WriteByte(Addr_GND,0x2E,0x11);//P2 NXT
   IS_IIC_WriteByte(Addr_GND,0x2F,0x01);//Loop Times

   //P3 time and cycle
   IS_IIC_WriteByte(Addr_GND,0x3D,0x15);//Color Cycle Times
   IS_IIC_WriteByte(Addr_GND,0x3C,0x07);//P4 Color Enable
   IS_IIC_WriteByte(Addr_GND,0x39,0x44);//TS &T1 Setting
   IS_IIC_WriteByte(Addr_GND,0x3A,0x44);//T2 &T3 Setting
   IS_IIC_WriteByte(Addr_GND,0x3B,0x44);//TP &T4 Setting
   IS_IIC_WriteByte(Addr_GND,0x3E,0x11);//P3 NXT
   IS_IIC_WriteByte(Addr_GND,0x3F,0x01);//Loop Times

   //P4 time and cycle
   IS_IIC_WriteByte(Addr_GND,0x4D,0x15);//Color Cycle Times
   IS_IIC_WriteByte(Addr_GND,0x4C,0x07);//P4 Color Enable
   IS_IIC_WriteByte(Addr_GND,0x49,0x44);//TS &T1 Setting
   IS_IIC_WriteByte(Addr_GND,0x4A,0x44);//T2 &T3 Setting
   IS_IIC_WriteByte(Addr_GND,0x4B,0x44);//TP &T4 Setting
```

```
    IS_IIC_WriteByte(Addr_GND,0x4E,0x11);//P4 NXT
    IS_IIC_WriteByte(Addr_GND,0x4F,0x01);//Loop Times

    //out1 color
    IS_IIC_WriteByte(Addr_GND,0x10,0x01);
    IS_IIC_WriteByte(Addr_GND,0x13,0x01);
    IS_IIC_WriteByte(Addr_GND,0x16,0x01);
    IS_IIC_WriteByte(Addr_GND,0x20,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x23,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x26,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x30,0xff);
    IS_IIC_WriteByte(Addr_GND,0x33,0xff);
    IS_IIC_WriteByte(Addr_GND,0x36,0xff);

    //out2 color
    IS_IIC_WriteByte(Addr_GND,0x11,0x01);
    IS_IIC_WriteByte(Addr_GND,0x14,0x01);
    IS_IIC_WriteByte(Addr_GND,0x17,0x01);
    IS_IIC_WriteByte(Addr_GND,0x21,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x24,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x27,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x31,0xff);
    IS_IIC_WriteByte(Addr_GND,0x34,0xff);
    IS_IIC_WriteByte(Addr_GND,0x37,0xff);

    //out3 color
    IS_IIC_WriteByte(Addr_GND,0x12,0x01);
    IS_IIC_WriteByte(Addr_GND,0x15,0x01);
    IS_IIC_WriteByte(Addr_GND,0x18,0x01);
    IS_IIC_WriteByte(Addr_GND,0x22,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x25,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x28,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x32,0xff);
    IS_IIC_WriteByte(Addr_GND,0x35,0xff);
    IS_IIC_WriteByte(Addr_GND,0x38,0xff);

    //out3 color
    IS_IIC_WriteByte(Addr_GND,0x40,0xff);
    IS_IIC_WriteByte(Addr_GND,0x41,0xff);
    IS_IIC_WriteByte(Addr_GND,0x42,0xff);
    IS_IIC_WriteByte(Addr_GND,0x43,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x44,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x45,0x3f);
    IS_IIC_WriteByte(Addr_GND,0x46,0x01);
    IS_IIC_WriteByte(Addr_GND,0x47,0x01);
    IS_IIC_WriteByte(Addr_GND,0x48,0x01);

    //updata
    IS_IIC_WriteByte(Addr_GND,0x50,0x00);
    IS_IIC_WriteByte(Addr_GND,0x51,0x00);
    IS_IIC_WriteByte(Addr_GND,0x52,0x00);
    IS_IIC_WriteByte(Addr_GND,0x53,0x00);
    IS_IIC_WriteByte(Addr_GND,0x54,0x00);
}
```