## DESCRIPTION

The IS31FL3761 is a general purpose 33 × n(n=12~1) LED Matrix programmed via 1MHz I2C or 12MHz SPI compatible interface. Each LED can be dimmed individually with 12-bit PWM data, and each color sink can have 8-bit DC scaling data which allows 4096 steps of linear PWM dimming and each color sink has 64 steps of DC current adjustable level. The output current of each channel is designed to be 30mA, which can be adjusted by 8-bit global control register and the SL registers. Proprietary algorithms are used in IS31FL3761 to minimize power bus noise caused by passive components on the power bus such as MLCC decoupling capacitor.

Additionally, each LED open and short state can be detected, IS31FL3761 stores the open or short information in Open/Short Registers. The Open/Short Registers can be read out via I2C/SPI compatible interface, inform MCU whether there are LEDs open or short and the locations of open or short LEDs.

The IS31FL3761 operates from 2.7V to 5.5V and features a very low shutdown and operational current.

IS31FL3761 is available in QFN-60 (7mm×7mm) package. It operates from 2.7V to 5.5V over the temperature range of -40°C to +125°C.

## QUICK START

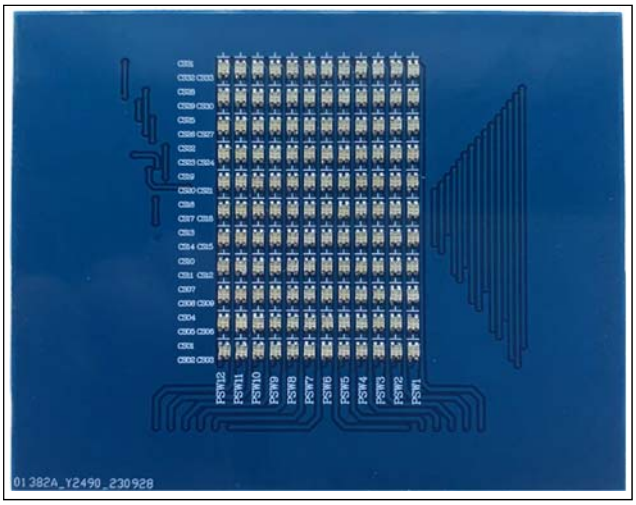


*Figure 1: Photo of IS31FL3761 Evaluation Board*

## FEATURES

- Supply voltage range: 2.7V to 5.5V
- Support 33 × n (n=12~1) matrix configuration
- Individual 12-bit/8+4-bit/8-bit/6+2-bit PWM control steps
- 26kHz frame rate (6+2bit PWM mode)
- Ultra-low Iq (0.59mA typ.) when frame rate is 333Hz
- Global 256 step current setting
- Each current sink (CSy) 8-bit DC current steps
- 1MHz I2C or 12MHz SPI by SEL pin
- For matrix scanning operation
  - Built-in de-ghosting circuit
  - Reduced inactive LED reverse bias to improve LED reliability
- LED open/short detection accessible to SPI/I2C
- Group phase shift (180 degree) to reduce audible noise and power ripple
- Spread spectrum
- ±6% device to device @$I_{OUT}$=30mA
- ±6% bit to bit @$I_{OUT}$=30mA
- QFN-60 (7mm×7mm) package
- RoHS & Halogen-Free Compliance
- TSCA Compliance

## 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 |
|---|---|---|
| IS31FL3761-QFLS4-EB | -40°C to +125°C, Industrial | QFN-60, 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 IS31FL3761 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 JP1 to enable the control of board MCU (default status).
2) Connect the 5VDC power to VCC (JP2) and GND (JP3), or plug in the USB power input to micro-USB.
3) Short the SEL and PVCC in CON6 to select the SPI interface.
4) Short the Pin 2 and Pin 3 in CON8 / CON9 / CON10 /CON11 to connect the SPI interface between IS31FL3761 and MCU.
5) Short the VIO and 3V in CON4 to keep the voltage reference same as MCU.
6) 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 IS31FL3761 evaluation board has five display modes. Press K1 to switch configurations:

***Note: See Appendix for each mode's detail.***

1) (Default mode) Rainbow #1.
2) Rainbow #2.
3) Mask
4) Rain.
5) White.

***Note: IS31FL3761 solely controls the FxLED function on the evaluation board.***

## SOFTWARE SUPPORT

JP1 default setting is close circuit. If it is set to open, the on-board MCU will stop working. The SPI pins and SDB pin are set to High Impedance. External SPI signals can be connected to in CON8 / CON9 / CON10 /CON11 and SDB signals can be connected to TP4 to control the IS31FL3761 LED driver.

The steps listed below are an example using the Arduino for 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 IS31FL3761 test firmware (sketch) from the Lumissil website http://www.lumissil.com/products/led-driver/fxled.

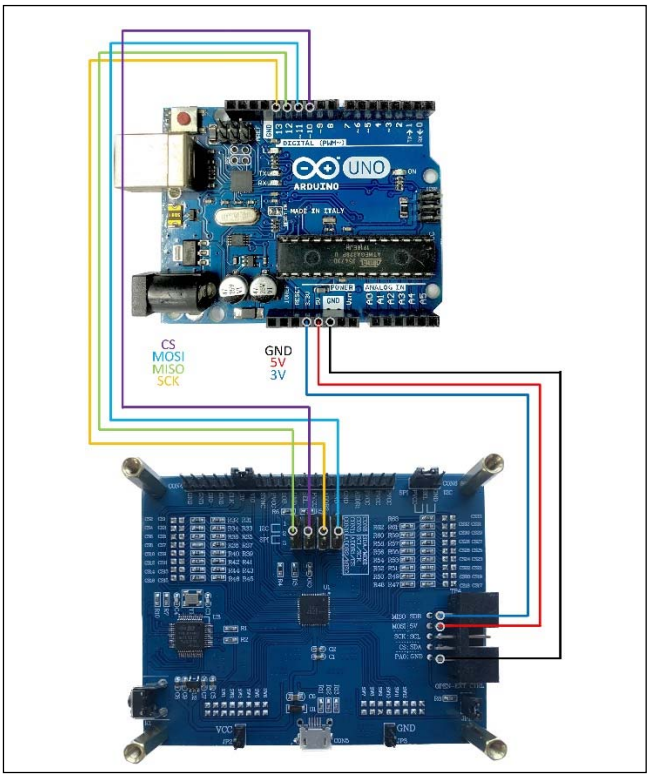


***Figure 2: Photo of Arduino UNO connected to Evaluation Board***

1) Open JP1.
2) Short the SEL and GND in CON6.
3) Open CON8/CON9/CON10/CON11.
4) Connect the 7 pins from Arduino board to IS31FL3761 EVB:
   a) Arduino 5V pin to IS31FL3761 EVB VCC in TP4.
   b) Arduino GND to IS31FL3761 EVB GND in TP4.
   c) Arduino 3.3V pin to IS31FL3761 EVB SDB in TP4.
   d) Arduino SCK (13) to IS31FL3761 EVB SCL/SCK (Pin 2 in CON9).
   e) Arduino MISO (12) to IS31FL3761 EVB ADDR2/MISO (Pin 2 in CON10).
   f) Arduino MOSI (11) to IS31FL3761 EVB SDA/MOSI (Pin 2 in CON8).
   g) Arduino SS (10) to IS31FL3761 EVB ADDR1/CS (Pin 2 in CON11).
   h) If Arduino use 3.3V MCU VCC, connect 3.3V to IS31FL3761 EVB SDB, if Arduino use 5.0V MCU VCC, connect 5.0V to EVB SDB. (Arduino UNO is 5.0V, so SDB=5.0V)
5) Use the test code in appendix Ⅰ or download the test firmware (sketch) from the Lumissil website, a .txt file and copy the code to Arduino IDE, compile and upload to Arduino.
6) Run the Arduino code and the initial mode will change the RGB LED colors.

***Please refer to the datasheet to get more information about IS31FL3761.***

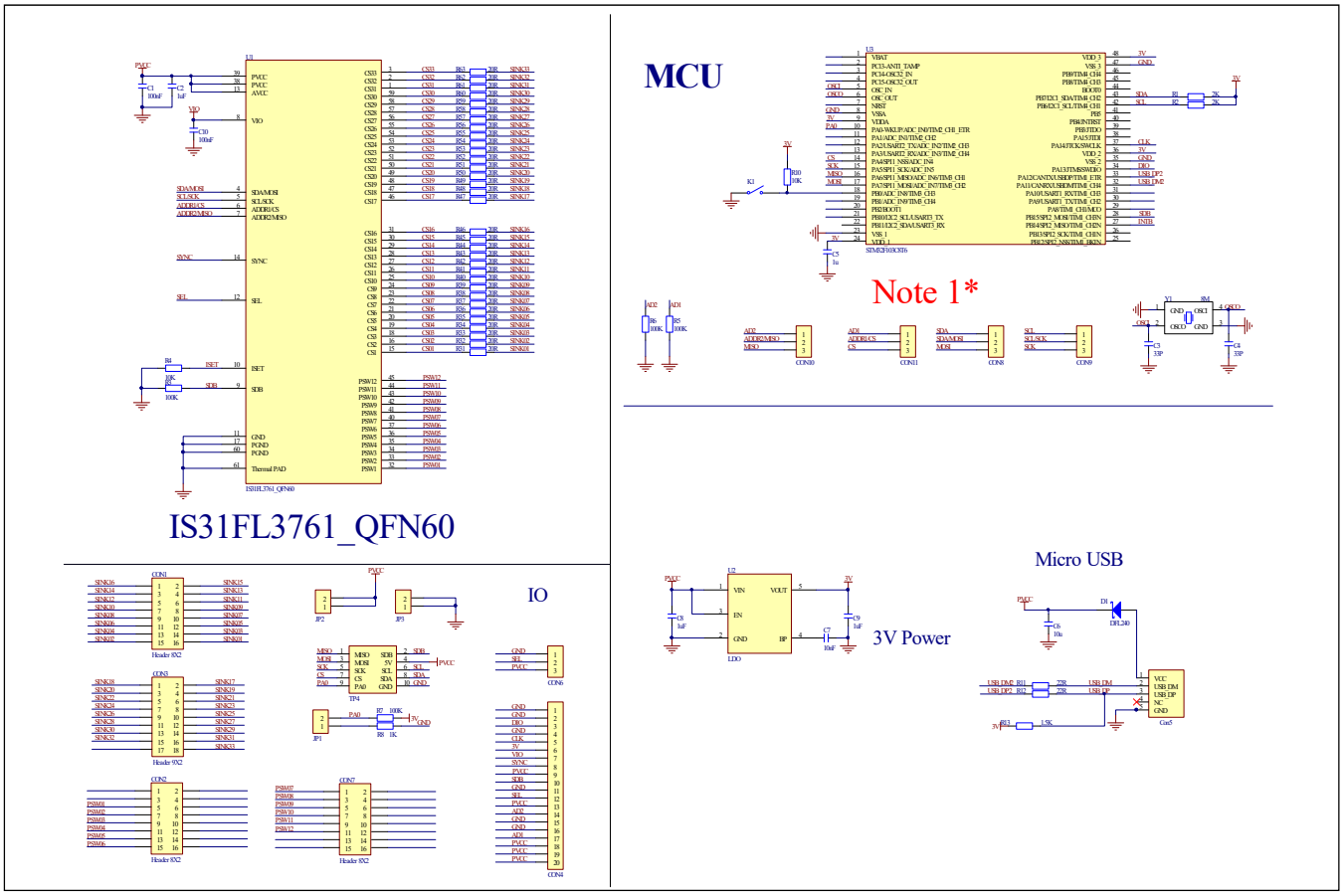


*Figure 3: IS31FL3761 Application Schematic*

***Note 1: IS31FL3761 EVB use SPI interface and CON10 (MISO), CON8 (MOSI), CON11 (CS), CON9 (SCK) need to short pin 2 and pin 3 by Jumpers, and the SEL pin need connect to PVCC in CON6. In order to use the I2C interface, CON10 (AD2), CON8 (SDA), CON11 (AD1), CON9 (SCL) need to short pin 2 and pin 1 by Jumpers, and the SEL pin need connect to GND in CON6.***

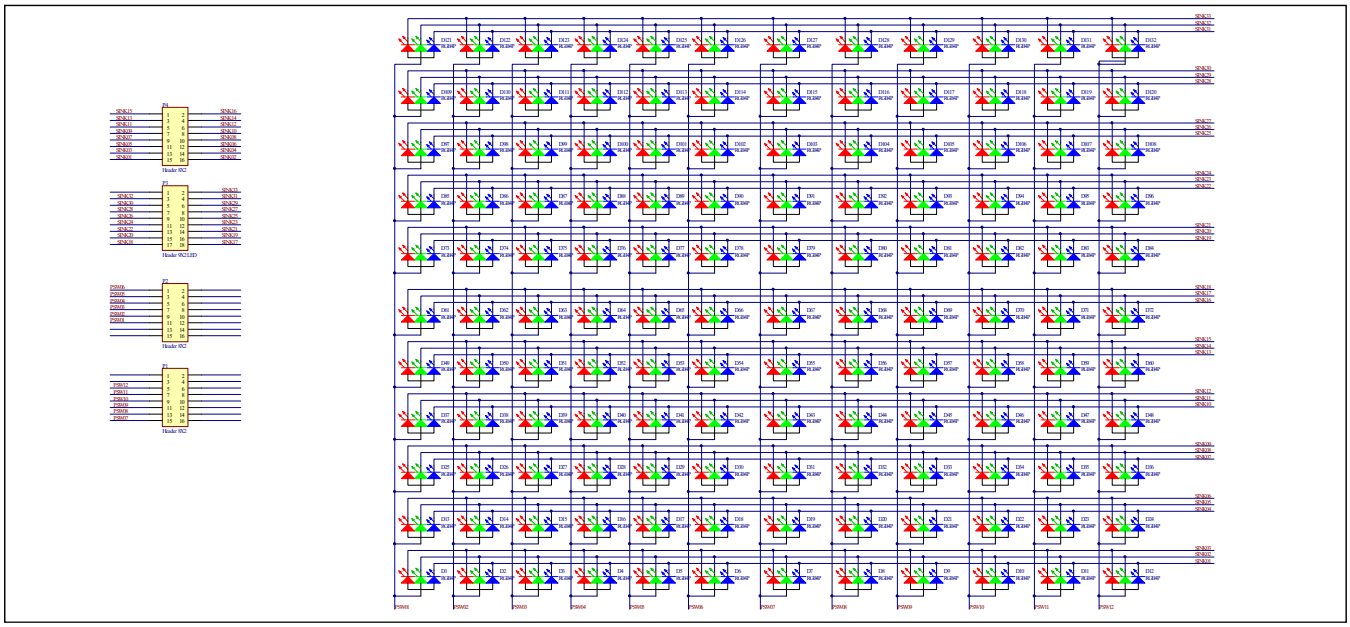

*Figure 4: FxLED RGB 12x11 ARRAY Application Schematic*

## BILL OF MATERIALS

### IS31FL3761

| Name | Symbol | Description | Qty | Supplier | Part No. |
|---|---|---|---|---|---|
| LED Driver | U1 | Matrix LED Driver | 1 | Lumissil | IS31FL3761 |
| LDO | U2 | Reduced voltage | 1 | SGMICRO | SGM2019-3.3V |
| MCU | U3 | Microcontroller | 1 | STM | STM32F103C8T6 |
| Diode | D1 | Diode, SMD | 1 | DIODES | DFLS240 |
| Crystal | Y1 | Crystal, 8MHz | 1 | JB | HC-49S |
| Resistor | R1, R2 | RES,2k,1/10W, ±5%, SMD | 2 | Yageo | RC0603JR-074K7L |
| Resistor | R3, R5, R6, R7 | RES,100k,1/10W, ±5%, SMD | 4 | Yageo | RC0603JR-07100KL |
| Resistor | R4, R10 | RES,10k,1/10W, ±5%, SMD | 2 | Yageo | RC0603JR-0710KL |
| Resistor | R8 | RES,1k,1/10W, ±5%, SMD | 1 | Yageo | RC0603JR-071KL |
| Resistor | R11, R12 | RES,22R,1/10W, ±5%, SMD | 2 | Yageo | RC0603JR-0722RL |
| Resistor | R13 | RES,1.5k,1/10W, ±5%, SMD | 1 | Yageo | RC0603JR-071K5L |
| Resistor | R31,R32,R34, R35,~R58, R58,R61,R62 | RES,20R,1/10W, ±5%, SMD (Note 2) | 22 | Yageo | RC0603JR-0720RL |
| Resistor | R33,R36~R60, R63 | RES,20R,1/10W, ±5%, SMD (Note 2) | 11 | Yageo | RC0603JR-0720RL |
| Capacitor | C1, C10 | CAP,100nF,16V, ±20%, SMD | 2 | Yageo | CC0603MRX7R7BB104 |
| Capacitor | C2, C5, C8, C9 | CAP, 1μF,16V, ±10%, SMD | 4 | Yageo | CC0603KRX7R7BB105 |
| Capacitor | C3, C4 | CAP,33pF,50V, ±5%, SMD | 2 | Yageo | CQ0603JRNPO9BN330 |
| Capacitor | C6 | CAP,10μF,16V, ±20%, SMD | 1 | Yageo | CC0603MRX5R7BB106 |
| Capacitor | C7 | CAP,10nF,16V, ±10%, SMD | 1 | Yageo | CC0603KRX7R7BB103 |
| Button | K1 | Button SMD | 1 | | |

***Bill of Materials, refer to Figure 3 above.***

### FxLED RGB 12x11 ARRAY

| Name | Symbol | Description | Qty | Supplier | Part No. |
|---|---|---|---|---|---|
| Diode | D1~D132 | RGB LED, SMD | 132 | Everlight | 9-237/R6GHBHC-A01/2T |

***Bill of Materials, refer to Figure 4 above.***

***Note 1: IS31FL3761 EVB use SPI interface and CON10 (MISO), CON8 (MOSI), CON11 (CS), CON9 (SCK) need to short pin 2 and pin 3 by Jumpers, and the SEL pin need connect to PVCC in CON6. In order to use the I2C interface, CON10 (AD2), CON8 (SDA), CON11 (AD1), CON9 (SCL) need to short pin 2 and pin 1 by Jumpers, and the SEL pin need connect to GND in CON6.***

***Note 2: The value of these resistors on the evaluation board is 20Ω. For $PV_{CC}$=5V and red LED application, prefer 51Ω for these resistors as shown in datasheet Figure 1.***

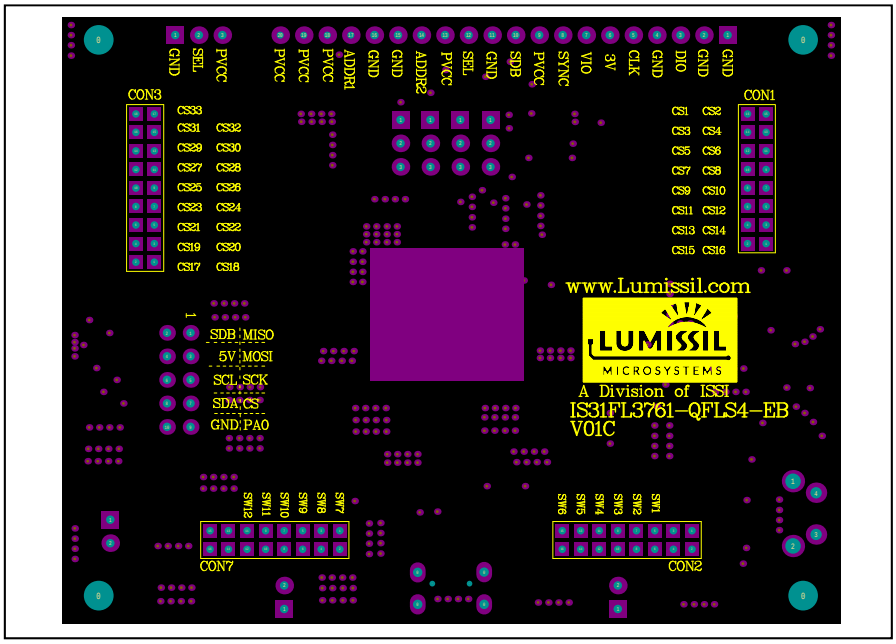


*Figure 5: Board Component Placement Guide - Top Layer*

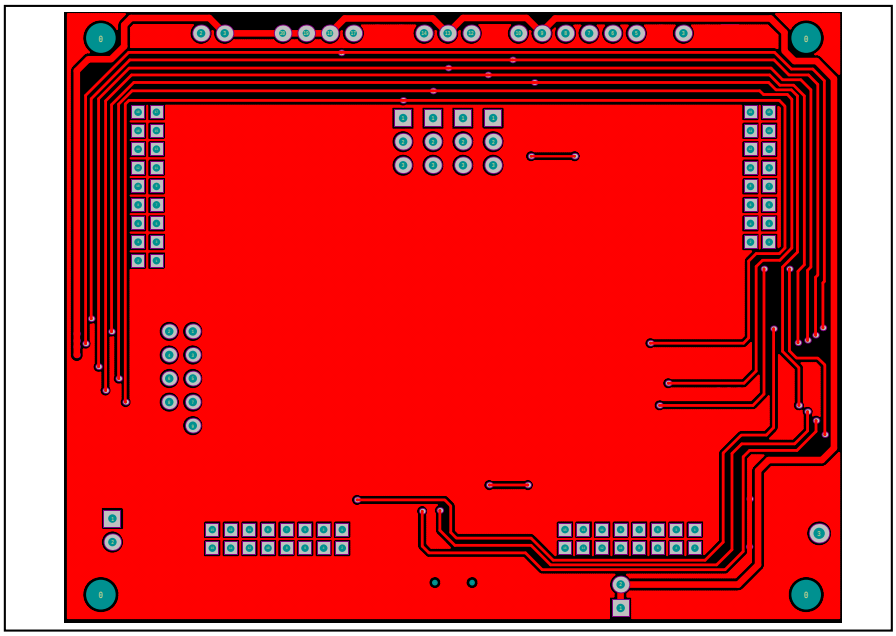

*Figure 6: Board PCB Layout - Top Layer*

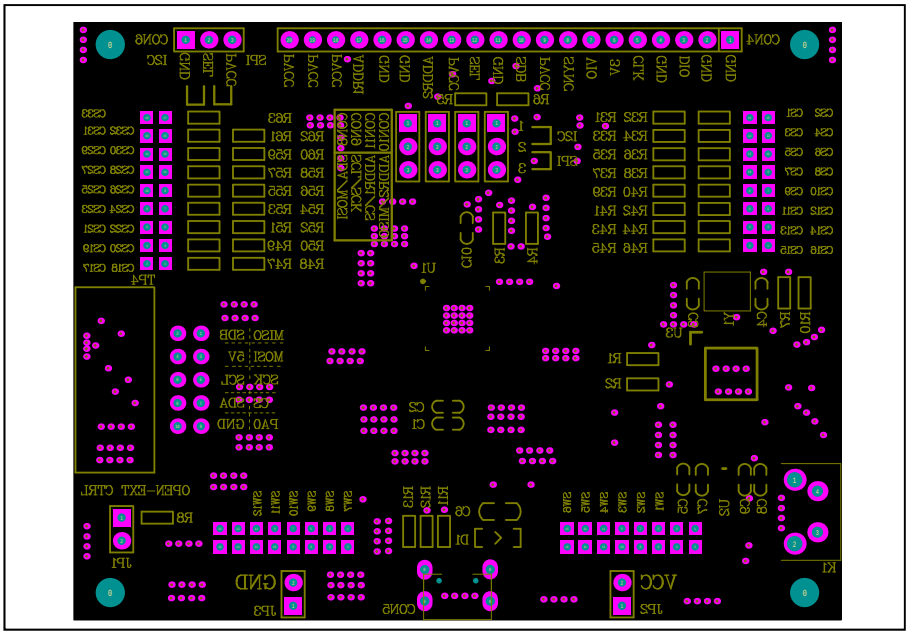

*Figure 7: Board Component Placement Guide - Bottom Layer*

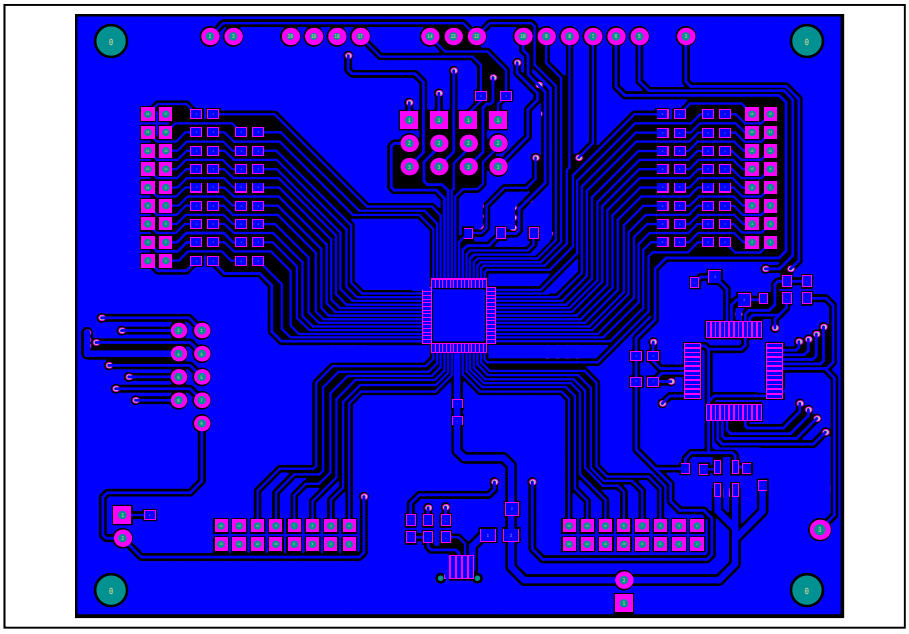

*Figure 8: Board PCB Layout - Bottom Layer*

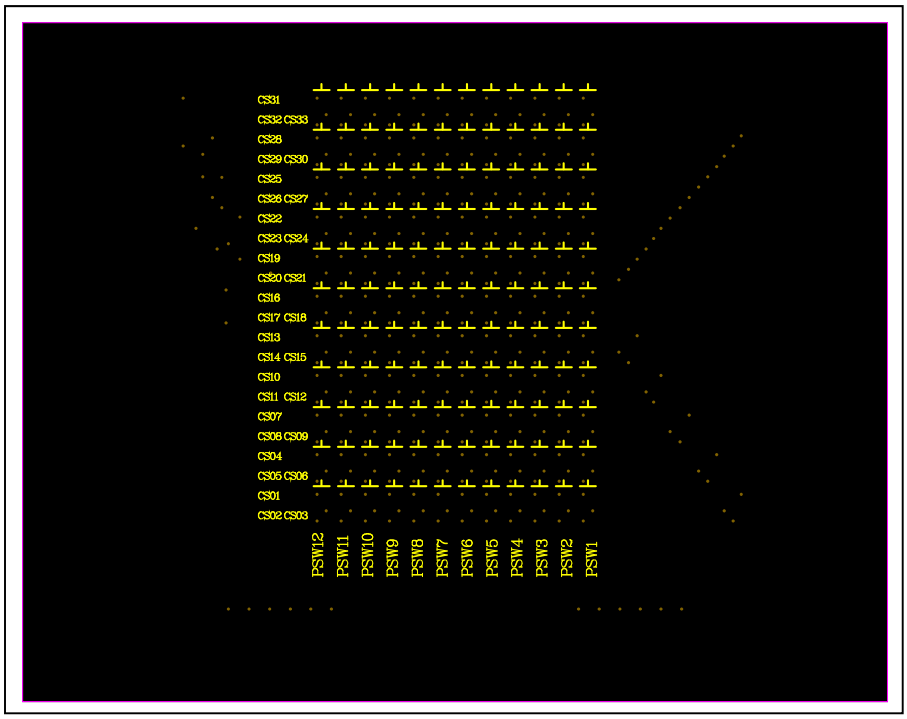


*Figure 9: Board Component Placement Guide - Top Layer*

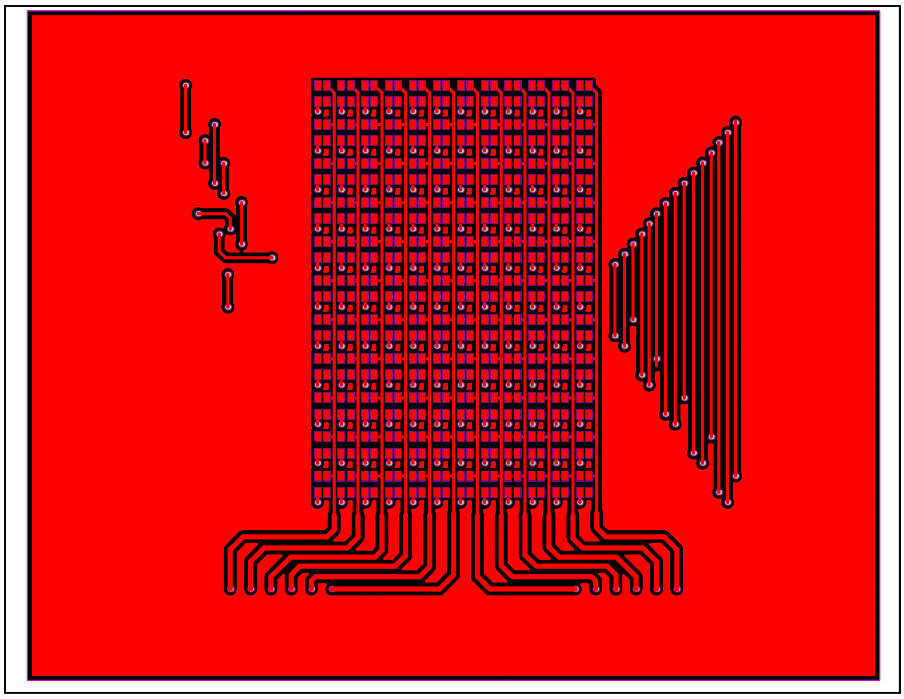

*Figure 10: Board PCB Layout - Top Layer*

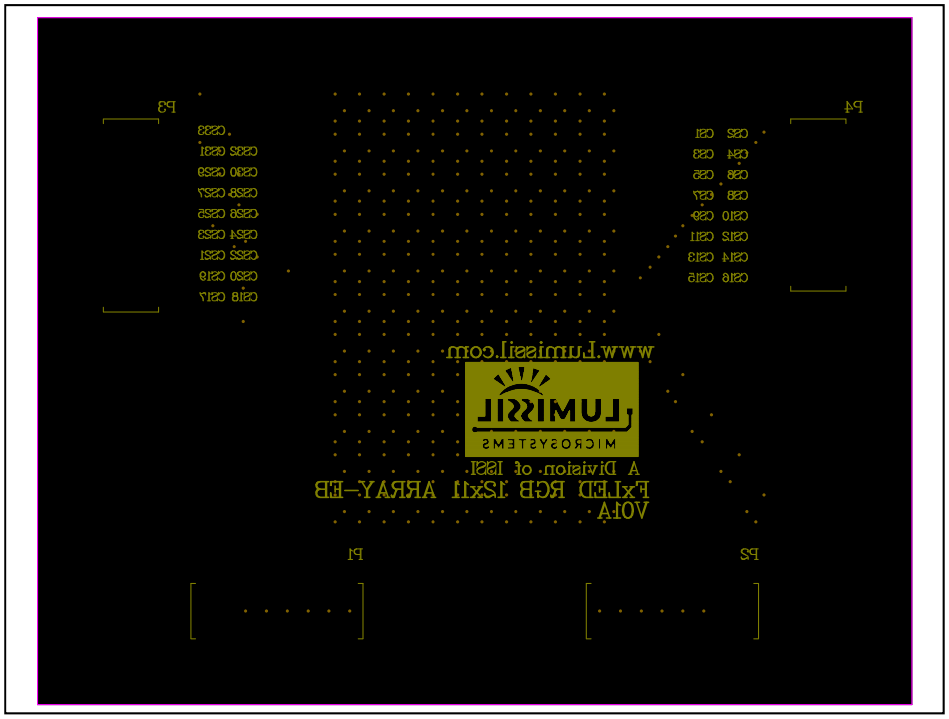

*Figure 11: Board Component Placement Guide - Top Layer*

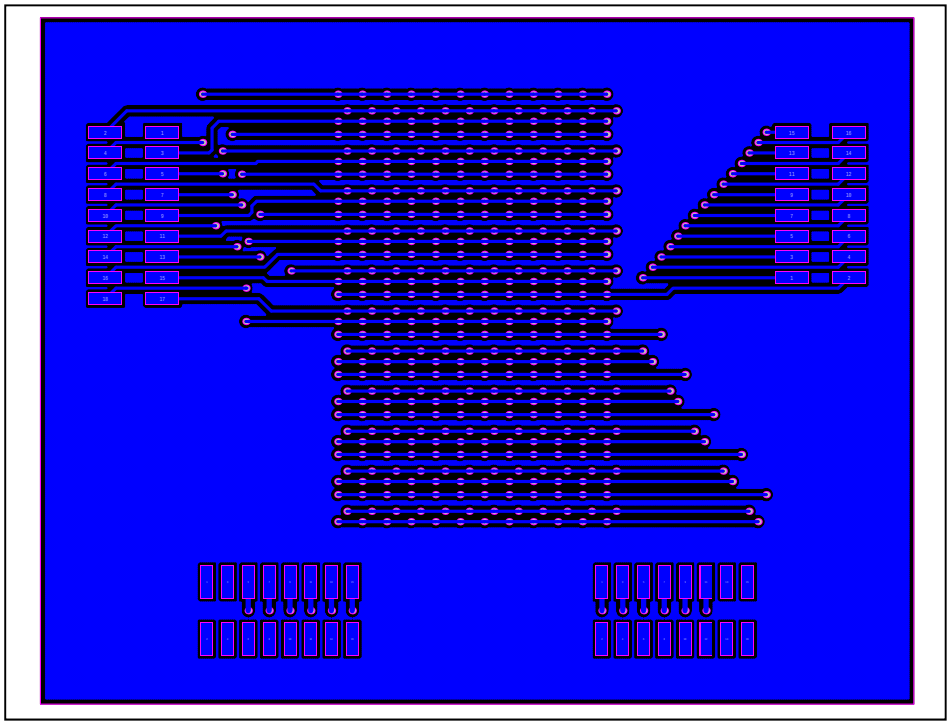

*Figure 12: Board PCB Layout - Top Layer*

## REVISION HISTORY

| Revision | Detail Information | Date |
|---|---|---|
| A | Initial release | 2023.10.16 |

## APPENDIX Ⅰ: IS31FL3761 Arduino Test Code V01A

```
#include<SPI.h>
#include<avr/pgmspace.h>

#define Addr_Write_Page0 0x30
#define Addr_Write_Page1 0x31
#define Addr_Write_Page2 0x32
const int slaveSelectPin = 10;

byte PWM_Gamma64[64]=
{
   0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,
   0x08,0x09,0x0b,0x0d,0x0f,0x11,0x13,0x16,
   0x1a,0x1c,0x1d,0x1f,0x22,0x25,0x28,0x2e,
   0x34,0x38,0x3c,0x40,0x44,0x48,0x4b,0x4f,
   0x55,0x5a,0x5f,0x64,0x69,0x6d,0x72,0x77,
   0x7d,0x80,0x88,0x8d,0x94,0x9a,0xa0,0xa7,
   0xac,0xb0,0xb9,0xbf,0xc6,0xcb,0xcf,0xd6,
   0xe1,0xe9,0xed,0xf1,0xf6,0xfa,0xfe,0xff
};


void setup()
{
   // put your setup code here, to run once:
   // set the slaveSelectPin as an output:
   pinMode (slaveSelectPin, OUTPUT);
   // initialize SPI:
   SPI.begin();
   SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE0));
//   SPI.setClockDivider(SPI_CLOCK_DIV4);
   SPI.setDataMode(3);

    Init3761();
}

void loop() {
   // put your main code here, to run repeatedly:
   IS31FL3761_MODE1();
}

void SPI_WriteByte(uint8_t Dev_Add,uint8_t Reg_Add,uint8_t Reg_Dat)   //writing an LED register
{
   digitalWrite(slaveSelectPin, LOW);     // take the SS pin low to select the chip:
   SPI.transfer(Dev_Add);     //    send in the address and value via SPI:
   SPI.transfer(Reg_Add);
   SPI.transfer(Reg_Dat);
   digitalWrite(slaveSelectPin, HIGH);     // take the SS pin high to de-select the chip:
}

void SPI_WriteBuffer(uint8_t pBuffer, int length, uint8_t Dev_Add,uint8_t Reg_Add)   //writing an LED register (Automatic address increment)
{
   digitalWrite(slaveSelectPin, LOW);     // take the SS pin low to select the chip:
   SPI.transfer(Dev_Add);     //    send in the address and value via SPI:
   SPI.transfer(Reg_Add);

   while(length--)
   {
      SPI.transfer(pBuffer);
   }
   digitalWrite(slaveSelectPin, HIGH);     // take the SS pin high to de-select the chip:
}

uint8_t SPI_ReadByte(uint8_t address)
{
   digitalWrite(slaveSelectPin,LOW);
   SPI.transfer(0x05);
   SPI.transfer(address);
   uint8_t result = SPI.transfer(0x00);
   digitalWrite(slaveSelectPin, HIGH);
   return result;
 }
void Init3761(void)
```

```
{
   int i,j;
   for(i=0x01;i<=0xFC;i++)
   {
      SPI_WriteByte(Addr_Write_Page0,i,0X00);//PAGE0 PWM
   }

      for(i=0x01;i<=0xFC;i++)
   {
      SPI_WriteByte(Addr_Write_Page1,i,0X00);//PAGE1 PWM
   }

      for(i=0x01;i<=0x5A;i++)
   {
      SPI_WriteByte(Addr_Write_Page2,i,0X00);//PAGE2 PWM
   }

   for(i=0x5B;i<=0x9C;i++)
   {
      SPI_WriteByte(Addr_Write_Page2,i,0xff);//PAGE2 scaling
   }

   SPI_WriteByte(Addr_Write_Page2,0xA0,0x01);//Configuration Register
   SPI_WriteByte(Addr_Write_Page2,0xA1,0x40);//GCC
   SPI_WriteByte(Addr_Write_Page0,0xFD,0x00);//PWM UPDATE
   delay(50);

}

void IS31FL3761_MODE1(void)//
{
   int i,j;
   while(1)
   {
      for(j=0;j<64;j++)// LED Breath Up
      {
         SPI_WriteBuffer(PWM_Gamma64[j], 0xFC, Addr_Write_Page0, 0x01);//PWM PAGE0
         SPI_WriteBuffer(PWM_Gamma64[j], 0XFC, Addr_Write_Page1, 0x01);//PWM PAGE1
         SPI_WriteBuffer(PWM_Gamma64[j], 0X5A, Addr_Write_Page2, 0x01);//PWM PAGE2
         SPI_WriteByte(Addr_Write_Page0,0xFD,0x00);//PWM UPDATE
       }

      for(j=63;j>=0;j--)// LED Breath Down
      {
         SPI_WriteBuffer(PWM_Gamma64[j], 0xFC, Addr_Write_Page0, 0x01);//PWM PAGE0
         SPI_WriteBuffer(PWM_Gamma64[j], 0XFC, Addr_Write_Page1, 0x01);//PWM PAGE1
         SPI_WriteBuffer(PWM_Gamma64[j], 0X5A, Addr_Write_Page2, 0x01);//PWM PAGE2
         SPI_WriteByte(Addr_Write_Page0,0xFD,0x00);//PWM UPDATE
       }
   }

}
```